

---

# **MPC5775E/MPC5775B Reference Manual**

Document Number: MPC5775E  
Rev. 1, 05/2018





## Contents

| Section number            | Title                        | Page |
|---------------------------|------------------------------|------|
| <b>Chapter 1</b>          |                              |      |
| <b>Preface</b>            |                              |      |
| 1.1                       | Overview.....                | 71   |
| 1.2                       | Device versions.....         | 71   |
| 1.3                       | Audience.....                | 71   |
| 1.4                       | Document organization.....   | 71   |
| 1.5                       | Conventions.....             | 72   |
| 1.5.1                     | Numbering systems.....       | 72   |
| 1.5.2                     | Typographic notation.....    | 72   |
| 1.5.3                     | Special terms.....           | 73   |
| 1.6                       | References.....              | 73   |
| <b>Chapter 2</b>          |                              |      |
| <b>Introduction</b>       |                              |      |
| 2.1                       | Overview.....                | 75   |
| 2.2                       | Device Description.....      | 75   |
| 2.2.1                     | Family comparison.....       | 75   |
| 2.2.2                     | Block diagram.....           | 77   |
| 2.2.3                     | Packages.....                | 78   |
| 2.2.4                     | Features summary.....        | 79   |
| <b>Chapter 3</b>          |                              |      |
| <b>Signal Description</b> |                              |      |
| 3.1                       | Production packages.....     | 81   |
| 3.2                       | Pad ring and ball maps.....  | 81   |
| 3.3                       | Pin function selection.....  | 81   |
| 3.3.1                     | Pad output multiplexing..... | 81   |
| 3.3.2                     | Pin input multiplexing.....  | 81   |
| 3.3.3                     | LVDS signal selection.....   | 82   |

| Section number | Title                                                           | Page |
|----------------|-----------------------------------------------------------------|------|
| 3.4            | External signal descriptions, multiplexing, and attributes..... | 83   |
| 3.4.1          | IBE and OBE control special cases.....                          | 83   |
| 3.4.2          | FCCU signals.....                                               | 90   |

## Chapter 4 Memory Map, Interrupts, and DMA Requests

|     |                                   |     |
|-----|-----------------------------------|-----|
| 4.1 | Memory map.....                   | 93  |
| 4.2 | Interrupt vector assignments..... | 100 |
| 4.3 | DMA request mapping.....          | 120 |

## Chapter 5 Device Configuration Format (DCF) Records

|        |                                                                                      |     |
|--------|--------------------------------------------------------------------------------------|-----|
| 5.1    | Overview.....                                                                        | 125 |
| 5.2    | DCF records in boot sequence.....                                                    | 126 |
| 5.3    | DCF records location.....                                                            | 126 |
| 5.4    | DCF record structure.....                                                            | 126 |
| 5.5    | DCF records sequence.....                                                            | 127 |
| 5.6    | DCF client Chip Select module assignments.....                                       | 129 |
| 5.7    | DCF clients.....                                                                     | 130 |
| 5.7.1  | UTEST Miscellaneous DCF client.....                                                  | 137 |
| 5.7.2  | UTEST Secure Boot Configurations DCF client.....                                     | 140 |
| 5.7.3  | UTEST Secure Boot Code Address DCF client.....                                       | 141 |
| 5.7.4  | UTEST Secure Boot Code Length DCF client.....                                        | 142 |
| 5.7.5  | UTEST Analog External Oscillator (XOSC ANA) DCF client.....                          | 143 |
| 5.7.6  | UTEST Error Functions Pads Configuration DCF client.....                             | 143 |
| 5.7.7  | UTEST Reset Escalation Configuration DCF client.....                                 | 145 |
| 5.7.8  | UTEST PMC Reset Enable Control (REE CTRL) DCF client.....                            | 146 |
| 5.7.9  | UTEST PMC Temperature Sensor Reset Event Enable Control (REE TMPSNS) DCF client..... | 146 |
| 5.7.10 | UTEST PMC Voltage Regulator Enable (VREG ENB) DCF client.....                        | 146 |

## Chapter 6 e200z7 Core Complex Overview

|     |                   |     |
|-----|-------------------|-----|
| 6.1 | Core complex..... | 149 |
|-----|-------------------|-----|



| Section number | Title                         | Page |
|----------------|-------------------------------|------|
| 6.2            | Core0 and Core1 overview..... | 149  |
| 6.3            | Checker core overview.....    | 149  |

## Chapter 7 Platform Coherency Unit (PCU)

|       |                                                                 |     |
|-------|-----------------------------------------------------------------|-----|
| 7.1   | Introduction.....                                               | 151 |
| 7.2   | Memory map and register descriptions.....                       | 151 |
| 7.2.1 | PCU Configuration and Error Status Register (PCU_CESR).....     | 152 |
| 7.2.2 | PCU Error Address Register, Core n (PCU_EAR <sub>n</sub> )..... | 156 |
| 7.2.3 | PCU Error Detail Register, Core n (PCU_EDR <sub>n</sub> ).....  | 157 |
| 7.2.4 | PCU Interrupt Register, Core n (PCU_IR <sub>n</sub> ).....      | 158 |

## Chapter 8 System Integration Unit (SIU, SIU\_B)

|        |                                                         |     |
|--------|---------------------------------------------------------|-----|
| 8.1    | Introduction.....                                       | 161 |
| 8.1.1  | Block diagram.....                                      | 161 |
| 8.1.2  | Overview.....                                           | 162 |
| 8.1.3  | Modes of operation.....                                 | 163 |
| 8.2    | Memory Map and Register Definition.....                 | 164 |
| 8.2.1  | MCU Identification Register (SIU_MIDR).....             | 217 |
| 8.2.2  | Reset Status Register (SIU_RSR).....                    | 218 |
| 8.2.3  | System Reset Control Register (SIU_SRCR).....           | 221 |
| 8.2.4  | External IRQ Status Register (SIU_EISR).....            | 222 |
| 8.2.5  | DMA/Interrupt Request Enable Register (SIU_DIRER).....  | 225 |
| 8.2.6  | DMA/Interrupt Request Select Register (SIU_DIRSR).....  | 228 |
| 8.2.7  | Overrun Status Register (SIU_OSR).....                  | 230 |
| 8.2.8  | Overrun Request Enable Register (SIU_ORER).....         | 232 |
| 8.2.9  | IRQ Rising-Edge Event Enable Register (SIU_IREER).....  | 234 |
| 8.2.10 | IRQ Falling-Edge Event Enable Register (SIU_IFEER)..... | 236 |
| 8.2.11 | IRQ Digital Filter Register (SIU_IDFR).....             | 239 |
| 8.2.12 | IRQ Filtered Input Register (SIU_IFIR).....             | 239 |

| Section number | Title                                                                          | Page |
|----------------|--------------------------------------------------------------------------------|------|
| 8.2.13         | Pad Configuration Register (SIU_PCR $n$ ).....                                 | 241  |
| 8.2.14         | GPIO Pin Data Output Register (SIU_GPDO $n$ ).....                             | 244  |
| 8.2.15         | GPIO Pin Data Input Register (legacy) (SIU_GPDInL).....                        | 245  |
| 8.2.16         | External IRQ Input Select Register (SIU_EIISR).....                            | 246  |
| 8.2.17         | DSPI Input Select Register (SIU_DISR).....                                     | 248  |
| 8.2.18         | eQADC Command FIFO Trigger Source Registers (SIU_ISEL4).....                   | 250  |
| 8.2.19         | eQADC Command FIFO Trigger Source Registers (SIU_ISEL5).....                   | 255  |
| 8.2.20         | eQADC Command FIFO Trigger Source Registers (SIU_ISEL6).....                   | 258  |
| 8.2.21         | eQADC Command FIFO Trigger Source Registers (SIU_ISEL7).....                   | 262  |
| 8.2.22         | eTPU_A Input Select Register (SIU_ISEL8).....                                  | 265  |
| 8.2.23         | eQADC Advance Trigger Source Register (SIU_ISEL9).....                         | 266  |
| 8.2.24         | Sigma-Delta Gate Selection Register (SIU_SDGATE_SEL).....                      | 269  |
| 8.2.25         | Sigma-Delta to eTPU_A Connection Configuration Register (SIU_SDETPUA).....     | 270  |
| 8.2.26         | Sigma-Delta to eTPU_B Connection Configuration Register (SIU_SDETPUB).....     | 273  |
| 8.2.27         | Sigma-Delta to eTPU_C Connection Configuration Register (SIU_SDETPUC).....     | 275  |
| 8.2.28         | Trigger Burst Generator Control Registers for eQADC_A (SIU_TBG_CR_A $n$ )..... | 278  |
| 8.2.29         | Trigger Burst Generator Control Registers for eQADC_B (SIU_TBG_CR_B $n$ )..... | 279  |
| 8.2.30         | External Clock Control Register (SIU_ECCR).....                                | 280  |
| 8.2.31         | System Clock Register (SIU_SYSDIV).....                                        | 281  |
| 8.2.32         | Halt Register 1 (SIU_HLT1).....                                                | 284  |
| 8.2.33         | Halt Acknowledge Register 1 (SIU_HLTACK1).....                                 | 288  |
| 8.2.34         | Core0 Reset Vector Register (SIU_RSTVEC0).....                                 | 291  |
| 8.2.35         | Core1 Reset Vector Register (SIU_RSTVEC1).....                                 | 292  |
| 8.2.36         | Core0 PID mapping control register (SIU_C0PID).....                            | 294  |
| 8.2.37         | Core1 PID mapping control register (SIU_C1PID).....                            | 295  |
| 8.2.38         | Halt Register 2 (SIU_HLT2).....                                                | 296  |
| 8.2.39         | Halt Acknowledge Register 2 (SIU_HLTACK2).....                                 | 299  |
| 8.2.40         | Sigma-Delta Clock Configuration Register (SIU_SDCLKCFG).....                   | 301  |
| 8.2.41         | Reset Configuration Register (SIU_RCR).....                                    | 302  |

| Section number | Title                                                                      | Page |
|----------------|----------------------------------------------------------------------------|------|
| 8.2.42         | Lock-Step Status Register (SIU_LOCKSTEP).....                              | 302  |
| 8.2.43         | Progressive Clock Switch Interrupt Enable Register (SIU_PCSER).....        | 303  |
| 8.2.44         | Progressive Clock Switch Interrupt Flag Register (SIU_PCSIFR).....         | 304  |
| 8.2.45         | Fast Ethernet Controller Control Register (SIU_FECCR).....                 | 305  |
| 8.2.46         | ECC Error Injection Register (SIU_ECCEIR).....                             | 305  |
| 8.2.47         | Parallel GPIO Pin Data Output Registers (SIU_PGPDOn).....                  | 306  |
| 8.2.48         | Parallel GPIO Pin Data Input Registers (SIU_PGPDIn).....                   | 310  |
| 8.2.49         | Masked Parallel GPIO Pin Data Output Registers (SIU_MPGPDOn).....          | 314  |
| 8.2.50         | Mask-Output High Register (SIU_DSPIInH).....                               | 318  |
| 8.2.51         | Mask-Output Low Register (SIU_DSPIInL).....                                | 322  |
| 8.2.52         | Serialized Output Signal Selection Register for DSPI_A (SIU_ETPUBA).....   | 327  |
| 8.2.53         | Serialized Output Signal Selection Register for DSPI_A (SIU_EMIOSA).....   | 331  |
| 8.2.54         | Serialized Output Signal Selection Register for DSPI_A (SIU_DSPIAHLA)..... | 334  |
| 8.2.55         | Serialized Output Signal Selection Register for DSPI_B (SIU_ETPUAB).....   | 338  |
| 8.2.56         | Serialized Output Signal Selection Register for DSPI_B (SIU_EMIOSB).....   | 342  |
| 8.2.57         | Serialized Output Signal Selection Register for DSPI_B (SIU_DSPIBHLB)..... | 346  |
| 8.2.58         | Serialized Output Signal Selection Register for DSPI_C (SIU_ETPUAC).....   | 350  |
| 8.2.59         | Serialized Output Signal Selection Register for DSPI_C (SIU_EMIOSC).....   | 353  |
| 8.2.60         | Serialized Output Signal Selection Register for DSPI_C (SIU_DSPICHLA)..... | 357  |
| 8.2.61         | Serialized Output Signal Selection Register for DSPI_C (SIU_ETPUBC).....   | 361  |
| 8.2.62         | Serialized Output Signal Selection Register for DSPI_D (SIU_ETPUBD).....   | 365  |
| 8.2.63         | Serialized Output Signal Selection Register for DSPI_D (SIU_EMIOSD).....   | 367  |
| 8.2.64         | Serialized Output Signal Selection Register for DSPI_D (SIU_DSPIDHLD)..... | 368  |
| 8.2.65         | GPIO Pin Data Input Register (SIU_GPDIn).....                              | 372  |
| 8.2.66         | Input Multiplexing Register0 (SIU_IMUX0).....                              | 373  |
| 8.2.67         | Input Multiplexing Register1 (SIU_IMUX1).....                              | 375  |
| 8.2.68         | Input Multiplexing Register2 (SIU_IMUX2).....                              | 376  |
| 8.2.69         | Input Multiplexing Register3 (SIU_IMUX3).....                              | 378  |
| 8.2.70         | Input Multiplexing Register4 (SIU_IMUX4).....                              | 379  |

| Section number | Title                                                                   | Page |
|----------------|-------------------------------------------------------------------------|------|
| 8.2.71         | Input Multiplexing Register5 (SIU_IMUX5).....                           | 381  |
| 8.2.72         | Input Multiplexing Register7 (SIU_IMUX7).....                           | 383  |
| 8.2.73         | Input Multiplexing Register10 (SIU_IMUX10).....                         | 385  |
| 8.2.74         | Input Multiplexing Register12 (SIU_IMUX12).....                         | 387  |
| 8.3            | SIU_B.....                                                              | 388  |
| 8.3.1          | Pad Configuration Register Mirrored (SIU_B_PCRnM).....                  | 389  |
| 8.3.2          | GPIO Pin Data Output Register Mirrored (SIU_B_GPDO <sub>n</sub> M)..... | 392  |
| 8.3.3          | GPIO Pin Data Input Register Mirrored (SIU_B_GPDI <sub>n</sub> M).....  | 393  |
| 8.4            | Functional Description.....                                             | 394  |
| 8.4.1          | Pad configuration.....                                                  | 394  |
| 8.4.2          | Interrupts.....                                                         | 394  |
| 8.4.3          | General Purpose I/O (GPIO).....                                         | 398  |
| 8.4.4          | Pin serialization for DSPI.....                                         | 398  |
| 8.4.5          | Internal multiplexing.....                                              | 400  |
| 8.4.6          | Trigger Burst Generators (TBG).....                                     | 401  |

## Chapter 9 Crossbar Switch (XBAR)

|       |                                                                     |     |
|-------|---------------------------------------------------------------------|-----|
| 9.1   | Chip-specific XBAR information.....                                 | 407 |
| 9.1.1 | XBAR master and slave assignments.....                              | 407 |
| 9.1.2 | Unimplemented PRS <sub>n</sub> and CRS <sub>n</sub> registers.....  | 408 |
| 9.1.3 | Reset value of PRS <sub>n</sub> and CRS <sub>n</sub> registers..... | 408 |
| 9.1.4 | Master priority elevation.....                                      | 409 |
| 9.2   | Introduction.....                                                   | 411 |
| 9.2.1 | Features.....                                                       | 411 |
| 9.3   | Memory Map / Register Definition.....                               | 411 |
| 9.3.1 | Priority Registers Slave (XBAR_PRS <sub>n</sub> ).....              | 412 |
| 9.3.2 | Control Register (XBAR_CRS <sub>n</sub> ).....                      | 415 |
| 9.4   | Functional Description.....                                         | 418 |
| 9.4.1 | General operation.....                                              | 418 |

| Section number | Title                                       | Page |
|----------------|---------------------------------------------|------|
| 9.4.2          | Register coherency.....                     | 419  |
| 9.4.3          | Arbitration.....                            | 419  |
| 9.5            | Initialization/application information..... | 421  |

## Chapter 10 Crossbar Integrity Checker (XBIC)

|        |                                               |     |
|--------|-----------------------------------------------|-----|
| 10.1   | Chip-specific XBIC information.....           | 423 |
| 10.1.1 | XBIC master and slave assignments.....        | 423 |
| 10.1.2 | Unimplemented MCR and ESR fields.....         | 423 |
| 10.2   | Overview.....                                 | 423 |
| 10.3   | Features.....                                 | 423 |
| 10.4   | Block diagram.....                            | 424 |
| 10.5   | External signal description.....              | 424 |
| 10.6   | Memory map and register definition.....       | 425 |
| 10.6.1 | XBIC Module Control Register (XBIC_MCR).....  | 425 |
| 10.6.2 | XBIC Error Injection Register (XBIC_EIR)..... | 427 |
| 10.6.3 | XBIC Error Status Register (XBIC_ESR).....    | 428 |
| 10.6.4 | XBIC Error Address Register (XBIC_EAR).....   | 430 |
| 10.7   | Functional description.....                   | 431 |

## Chapter 11 Peripheral Bridge (AIPS/PBRIDGE)

|        |                                                  |     |
|--------|--------------------------------------------------|-----|
| 11.1   | Chip-specific AIPS/PBRIDGE information.....      | 433 |
| 11.1.1 | Modules assigned to PACRn and OPACRn fields..... | 433 |
| 11.1.2 | AIPS/PBRIDGE register reset values.....          | 436 |
| 11.2   | Introduction.....                                | 437 |
| 11.2.1 | Features.....                                    | 437 |
| 11.2.2 | General operation.....                           | 437 |
| 11.3   | Memory map/register definition.....              | 437 |
| 11.3.1 | Master Privilege Register A (AIPS_MPRA).....     | 438 |
| 11.3.2 | Master Privilege Register B (AIPS_MPRB).....     | 442 |

| Section number | Title                                                                           | Page |
|----------------|---------------------------------------------------------------------------------|------|
| 11.3.3         | Peripheral Access Control Register (AIPS_PACR <sub>n</sub> ).....               | 444  |
| 11.3.4         | Off-Platform Peripheral Access Control Register (AIPS_OPACR <sub>n</sub> )..... | 449  |
| 11.4           | Functional description.....                                                     | 454  |
| 11.4.1         | Access support.....                                                             | 454  |

## Chapter 12 Platform Configuration Module (PCM)

|        |                                                                  |     |
|--------|------------------------------------------------------------------|-----|
| 12.1   | PCM memory map and register descriptions.....                    | 455 |
| 12.1.1 | FEC Burst Optimization Master Control Register (PCM_FBOMCR)..... | 456 |
| 12.1.2 | Bus Bridge Configuration Register 1 (PCM_IAHB_BE1).....          | 457 |
| 12.1.3 | Bus Bridge Configuration Register 2 (PCM_IAHB_BE2).....          | 460 |

## Chapter 13 Semaphores (SEMA4)

|         |                                                 |     |
|---------|-------------------------------------------------|-----|
| 13.1    | Introduction .....                              | 463 |
| 13.1.1  | Multi-Core Programming 101: Software Gates..... | 463 |
| 13.1.2  | Overview.....                                   | 465 |
| 13.1.3  | Features.....                                   | 466 |
| 13.1.4  | Modes of Operation.....                         | 467 |
| 13.2    | External Signal Description.....                | 467 |
| 13.3    | Memory map and register definition.....         | 468 |
| 13.3.1  | Semaphores Gate 0 Register (SEMA4_Gate00).....  | 469 |
| 13.3.2  | Semaphores Gate 1 Register (SEMA4_Gate01).....  | 470 |
| 13.3.3  | Semaphores Gate 2 Register (SEMA4_Gate02).....  | 471 |
| 13.3.4  | Semaphores Gate 3 Register (SEMA4_Gate03).....  | 472 |
| 13.3.5  | Semaphores Gate 4 Register (SEMA4_Gate04).....  | 473 |
| 13.3.6  | Semaphores Gate 5 Register (SEMA4_Gate05).....  | 474 |
| 13.3.7  | Semaphores Gate 6 Register (SEMA4_Gate06).....  | 475 |
| 13.3.8  | Semaphores Gate 7 Register (SEMA4_Gate07).....  | 476 |
| 13.3.9  | Semaphores Gate 8 Register (SEMA4_Gate08).....  | 477 |
| 13.3.10 | Semaphores Gate 9 Register (SEMA4_Gate09).....  | 478 |

| Section number | Title                                                              | Page |
|----------------|--------------------------------------------------------------------|------|
| 13.3.11        | Semaphores Gate 10 Register (SEMA4_Gate10).....                    | 479  |
| 13.3.12        | Semaphores Gate 11 Register (SEMA4_Gate11).....                    | 480  |
| 13.3.13        | Semaphores Gate 12 Register (SEMA4_Gate12).....                    | 481  |
| 13.3.14        | Semaphores Gate 13 Register (SEMA4_Gate13).....                    | 482  |
| 13.3.15        | Semaphores Gate 14 Register (SEMA4_Gate14).....                    | 483  |
| 13.3.16        | Semaphores Gate 15 Register (SEMA4_Gate15).....                    | 484  |
| 13.3.17        | Semaphores Processor n IRQ Notification Enable (SEMA4_CPnINE)..... | 485  |
| 13.3.18        | Semaphores Processor n IRQ Notification (SEMA4_CPnNTF).....        | 487  |
| 13.3.19        | Semaphores (Secure) Reset Gate n (SEMA4_RSTGT).....                | 488  |
| 13.3.20        | Semaphores (Secure) Reset IRQ Notification (SEMA4_RSTNTF).....     | 490  |
| 13.4           | Functional Description.....                                        | 491  |
| 13.4.1         | SEMA4_GATEn Operation.....                                         | 491  |
| 13.4.2         | SEMA4_CPnNTF Operation.....                                        | 493  |
| 13.5           | Initialization Information.....                                    | 496  |
| 13.6           | Application Information.....                                       | 496  |

## Chapter 14 Memory Protection Unit (MPU)

|        |                                                      |     |
|--------|------------------------------------------------------|-----|
| 14.1   | Chip-specific MPU information.....                   | 499 |
| 14.1.1 | MPU master and slave assignments.....                | 499 |
| 14.1.2 | MPU register reset values.....                       | 500 |
| 14.2   | Introduction.....                                    | 500 |
| 14.3   | Overview.....                                        | 500 |
| 14.3.1 | Block diagram.....                                   | 500 |
| 14.3.2 | Features.....                                        | 501 |
| 14.4   | Memory map/register definition.....                  | 502 |
| 14.4.1 | Control/Error Status Register (MPU_CESR).....        | 505 |
| 14.4.2 | Error Address Register, slave port n (MPU_EARn)..... | 506 |
| 14.4.3 | Error Detail Register, slave port n (MPU_EDRn).....  | 507 |
| 14.4.4 | Region Descriptor n, Word 0 (MPU_RGDn_WORD0).....    | 508 |

| Section number | Title                                                           | Page |
|----------------|-----------------------------------------------------------------|------|
| 14.4.5         | Region Descriptor n, Word 1 (MPU_RGDn_WORD1).....               | 508  |
| 14.4.6         | Region Descriptor n, Word 2 (MPU_RGDn_WORD2).....               | 509  |
| 14.4.7         | Region Descriptor n, Word 3 (MPU_RGDn_WORD3).....               | 512  |
| 14.4.8         | Region Descriptor Alternate Access Control n (MPU_RGDAACn)..... | 513  |
| 14.5           | Functional description.....                                     | 515  |
| 14.5.1         | Access evaluation macro.....                                    | 515  |
| 14.5.2         | Putting it all together and error terminations.....             | 517  |
| 14.5.3         | Power management.....                                           | 517  |
| 14.6           | Initialization information.....                                 | 518  |
| 14.7           | Application information.....                                    | 518  |

## Chapter 15 Interrupt Controller (INTC)

|        |                                                                                   |     |
|--------|-----------------------------------------------------------------------------------|-----|
| 15.1   | Chip-specific INTC information.....                                               | 521 |
| 15.2   | Introduction.....                                                                 | 521 |
| 15.3   | Block diagram.....                                                                | 522 |
| 15.4   | Features.....                                                                     | 524 |
| 15.5   | Modes of operation.....                                                           | 524 |
| 15.5.1 | Software vector mode.....                                                         | 524 |
| 15.5.2 | Hardware vector mode.....                                                         | 527 |
| 15.6   | External signal description.....                                                  | 528 |
| 15.7   | Memory map and register definition.....                                           | 528 |
| 15.7.1 | INTC Module Configuration Register (INTC_MCR).....                                | 543 |
| 15.7.2 | INTC Current Priority Register for Processor <i>n</i> (INTC_CPR_PRCn).....        | 544 |
| 15.7.3 | INTC Interrupt Acknowledge Register for Processor <i>n</i> (INTC_IACKR_PRCn)..... | 545 |
| 15.7.4 | INTC End Of Interrupt Register for Processor <i>n</i> (INTC_EOIR_PRCn).....       | 546 |
| 15.7.5 | INTC Software Set/Clear Interrupt Register (INTC_SSCIRn).....                     | 547 |
| 15.7.6 | INTC Priority Select Register (INTC_PSRn).....                                    | 548 |



| Section number | Title                                                              | Page |
|----------------|--------------------------------------------------------------------|------|
| 15.8           | Functional description.....                                        | 548  |
| 15.8.1         | Interrupt request sources.....                                     | 549  |
| 15.8.2         | Priority management.....                                           | 550  |
| 15.8.3         | Handshaking with processor.....                                    | 552  |
| 15.9           | Initialization/application information.....                        | 555  |
| 15.9.1         | Initialization flow.....                                           | 555  |
| 15.9.2         | Interrupt exception handler.....                                   | 556  |
| 15.9.3         | ISR, RTOS, and task hierarchy.....                                 | 557  |
| 15.9.4         | Order of execution.....                                            | 558  |
| 15.9.5         | Priority ceiling protocol.....                                     | 559  |
| 15.9.6         | Selecting priorities according to request rates and deadlines..... | 562  |
| 15.9.7         | Software-settable interrupt requests.....                          | 563  |
| 15.9.8         | Lowering priority within an ISR.....                               | 564  |
| 15.9.9         | Negating an interrupt request outside of its ISR.....              | 564  |
| 15.9.10        | Examining LIFO contents.....                                       | 565  |
| 15.10          | Interrupt sources.....                                             | 566  |

## Chapter 16

### Direct Memory Access Controller (eDMA)

|        |                                     |     |
|--------|-------------------------------------|-----|
| 16.1   | Chip-specific eDMA information..... | 567 |
| 16.2   | Introduction.....                   | 567 |
| 16.2.1 | eDMA system block diagram.....      | 567 |
| 16.2.2 | Block parts.....                    | 568 |
| 16.2.3 | Features.....                       | 569 |
| 16.3   | Modes of operation.....             | 570 |
| 16.4   | Memory map/register definition..... | 571 |
| 16.4.1 | TCD memory.....                     | 571 |
| 16.4.2 | TCD initialization.....             | 571 |
| 16.4.3 | TCD structure.....                  | 572 |
| 16.4.4 | Reserved memory and bit fields..... | 572 |

| Section number | Title                                                                                                                        | Page |
|----------------|------------------------------------------------------------------------------------------------------------------------------|------|
| 16.4.5         | Control Register (DMA_CR).....                                                                                               | 623  |
| 16.4.6         | Error Status Register (DMA_ES).....                                                                                          | 626  |
| 16.4.7         | Enable Request Register High (DMA_ERQH).....                                                                                 | 629  |
| 16.4.8         | Enable Request Register Low (DMA_ERQL).....                                                                                  | 632  |
| 16.4.9         | Enable Error Interrupt Register High (DMA_EEIH).....                                                                         | 636  |
| 16.4.10        | Enable Error Interrupt Register Low (DMA_EEIL).....                                                                          | 639  |
| 16.4.11        | Set Enable Request Register (DMA_SERQ).....                                                                                  | 643  |
| 16.4.12        | Clear Enable Request Register (DMA_CERQ).....                                                                                | 643  |
| 16.4.13        | Set Enable Error Interrupt Register (DMA_SEEI).....                                                                          | 644  |
| 16.4.14        | Clear Enable Error Interrupt Register (DMA_CEEI).....                                                                        | 645  |
| 16.4.15        | Clear Interrupt Request Register (DMA_CINT).....                                                                             | 646  |
| 16.4.16        | Clear Error Register (DMA_CERR).....                                                                                         | 647  |
| 16.4.17        | Set START Bit Register (DMA_SSRT).....                                                                                       | 648  |
| 16.4.18        | Clear DONE Status Bit Register (DMA_CDNE).....                                                                               | 649  |
| 16.4.19        | Interrupt Request Register High (DMA_INTH).....                                                                              | 649  |
| 16.4.20        | Interrupt Request Register Low (DMA_INTL).....                                                                               | 653  |
| 16.4.21        | Error Register High (DMA_ERRH).....                                                                                          | 657  |
| 16.4.22        | Error Register Low (DMA_ERRL).....                                                                                           | 661  |
| 16.4.23        | Hardware Request Status Register High (DMA_HRSH).....                                                                        | 664  |
| 16.4.24        | Hardware Request Status Register Low (DMA_HRSL).....                                                                         | 668  |
| 16.4.25        | Global PCU Output Register (DMA_GPOR <sub>n</sub> ).....                                                                     | 672  |
| 16.4.26        | Channel n Priority Register (DMA_DCHPRIn).....                                                                               | 673  |
| 16.4.27        | TCD Source Address (DMA_TCD <sub>n</sub> _SADDR).....                                                                        | 674  |
| 16.4.28        | TCD Transfer Attributes (DMA_TCD <sub>n</sub> _ATTR).....                                                                    | 674  |
| 16.4.29        | TCD Signed Source Address Offset (DMA_TCD <sub>n</sub> _SOFF).....                                                           | 675  |
| 16.4.30        | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD <sub>n</sub> _NBYTES_MLNO).....                                  | 675  |
| 16.4.31        | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled)<br>(DMA_TCD <sub>n</sub> _NBYTES_MLOFFNO)..... | 676  |

| Section number | Title                                                                                                       | Page |
|----------------|-------------------------------------------------------------------------------------------------------------|------|
| 16.4.32        | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled)<br>(DMA_TCDn_NBYTES_MLOFFYES).....     | 677  |
| 16.4.33        | TCD Last Source Address Adjustment (DMA_TCDn_SLAST).....                                                    | 678  |
| 16.4.34        | TCD Destination Address (DMA_TCDn_DADDR).....                                                               | 679  |
| 16.4.35        | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled)<br>(DMA_TCDn_CITER_ELINKYES).....   | 679  |
| 16.4.36        | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Disabled)<br>(DMA_TCDn_CITER_ELINKNO).....   | 680  |
| 16.4.37        | TCD Signed Destination Address Offset (DMA_TCDn_DOFF).....                                                  | 681  |
| 16.4.38        | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCDn_DLASTSGA).....                     | 682  |
| 16.4.39        | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled)<br>(DMA_TCDn_BITER_ELINKYES)..... | 682  |
| 16.4.40        | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled)<br>(DMA_TCDn_BITER_ELINKNO)..... | 683  |
| 16.4.41        | TCD Control and Status (DMA_TCDn_CSR).....                                                                  | 684  |
| 16.5           | Functional description.....                                                                                 | 686  |
| 16.5.1         | eDMA basic data flow.....                                                                                   | 686  |
| 16.5.2         | Fault reporting and handling.....                                                                           | 689  |
| 16.5.3         | Channel preemption.....                                                                                     | 692  |
| 16.5.4         | Performance.....                                                                                            | 692  |
| 16.6           | Initialization/application information.....                                                                 | 697  |
| 16.6.1         | eDMA initialization.....                                                                                    | 697  |
| 16.6.2         | Programming errors.....                                                                                     | 699  |
| 16.6.3         | Arbitration mode considerations.....                                                                        | 700  |
| 16.6.4         | Performing DMA transfers.....                                                                               | 702  |
| 16.6.5         | Monitoring transfer descriptor status.....                                                                  | 706  |
| 16.6.6         | Channel Linking.....                                                                                        | 708  |

| Section number | Title                                                                   | Page |
|----------------|-------------------------------------------------------------------------|------|
| 16.6.7         | Dynamic programming.....                                                | 709  |
| 16.6.8         | Suspend/resume a DMA channel with active hardware service requests..... | 712  |

## Chapter 17 Error Injection Module (EIM)

|        |                                                                   |     |
|--------|-------------------------------------------------------------------|-----|
| 17.1   | Chip-specific EIM information.....                                | 715 |
| 17.1.1 | EIM channels.....                                                 | 715 |
| 17.1.2 | EIM channel assignments.....                                      | 715 |
| 17.1.3 | EIM_EICHDN_Word register bit mapping.....                         | 716 |
| 17.2   | Introduction.....                                                 | 716 |
| 17.2.1 | Overview.....                                                     | 717 |
| 17.2.2 | Features.....                                                     | 718 |
| 17.3   | Memory map and register definition.....                           | 718 |
| 17.3.1 | Error Injection Module Configuration Register (EIM_EIMCR).....    | 720 |
| 17.3.2 | Error Injection Channel Enable register (EIM_EICHEN).....         | 721 |
| 17.3.3 | Error Injection Channel Descriptor, Word0 (EIM_EICHDN_WORD0)..... | 723 |
| 17.3.4 | Error Injection Channel Descriptor, Word1 (EIM_EICHDN_WORD1)..... | 724 |
| 17.3.5 | Error Injection Channel Descriptor, Word2 (EIM_EICHDN_WORD2)..... | 724 |
| 17.4   | Functional description.....                                       | 725 |
| 17.4.1 | Error injection scenarios.....                                    | 725 |

## Chapter 18 Error Reporting Module (ERM)

|        |                                                       |     |
|--------|-------------------------------------------------------|-----|
| 18.1   | Chip-specific ERM information.....                    | 727 |
| 18.1.1 | ERM memory channels.....                              | 727 |
| 18.1.2 | Sources of memory error events.....                   | 727 |
| 18.1.3 | Managing single-bit faults of TCD RAM for eDMA_A..... | 728 |
| 18.2   | Introduction.....                                     | 728 |
| 18.2.1 | Overview.....                                         | 728 |
| 18.2.2 | Features.....                                         | 729 |

| Section number | Title                                                                          | Page |
|----------------|--------------------------------------------------------------------------------|------|
| 18.3           | Memory map and register definition.....                                        | 729  |
| 18.3.1         | ERM Configuration Register (ERM_CR0).....                                      | 731  |
| 18.3.2         | ERM Configuration Register (ERM_CR1).....                                      | 734  |
| 18.3.3         | ERM Configuration Register (ERM_CR2).....                                      | 737  |
| 18.3.4         | ERM Status Register (ERM_SR0).....                                             | 740  |
| 18.3.5         | ERM Status Register (ERM_SR1).....                                             | 744  |
| 18.3.6         | ERM Status Register (ERM_SR2).....                                             | 748  |
| 18.3.7         | ERM Memory <i>n</i> Error Address Register (ERM_EAR <sub><i>n</i></sub> )..... | 750  |
| 18.3.8         | ERM Memory <i>n</i> Syndrome Register (ERM_SYN <sub><i>n</i></sub> ).....      | 751  |
| 18.4           | Functional description.....                                                    | 751  |
| 18.4.1         | Single-bit correction events.....                                              | 751  |
| 18.4.2         | Non-correctable error events.....                                              | 752  |
| 18.5           | Initialization.....                                                            | 753  |

## Chapter 19 Clocking

|        |                                    |     |
|--------|------------------------------------|-----|
| 19.1   | Introduction.....                  | 755 |
| 19.2   | Clock generation.....              | 756 |
| 19.3   | SIU registers.....                 | 757 |
| 19.4   | Operating frequencies.....         | 758 |
| 19.5   | Default clock configuration.....   | 759 |
| 19.6   | Clock sources.....                 | 760 |
| 19.6.1 | 16 MHz RC oscillator (IRC).....    | 760 |
| 19.6.2 | External oscillator (XOSC).....    | 760 |
| 19.6.3 | PLL.....                           | 761 |
| 19.7   | Clock gating.....                  | 764 |
| 19.8   | Core and on-platform clocking..... | 765 |
| 19.9   | Peripheral clocks.....             | 766 |
| 19.9.1 | eTPU clocking.....                 | 767 |
| 19.9.2 | FlexCAN and MCAN clocking.....     | 768 |

| Section number | Title                                     | Page |
|----------------|-------------------------------------------|------|
| 19.9.3         | Ethernet clocking.....                    | 768  |
| 19.9.4         | SDADC clocking.....                       | 768  |
| 19.10          | Serial boot clocking.....                 | 769  |
| 19.11          | MCKO.....                                 | 769  |
| 19.12          | CLKOUT / ENGCLK.....                      | 769  |
| 19.13          | Clock monitoring.....                     | 770  |
| 19.13.1        | CMU-monitored clocks.....                 | 770  |
| 19.13.2        | CMU_0 configuration.....                  | 770  |
| 19.13.3        | CMU_1 to CMU_8 configuration.....         | 771  |
| 19.13.4        | CMU output signals to FCCU and INTC.....  | 773  |
| 19.13.5        | CMU register and field availability ..... | 773  |
| 19.13.6        | CMU_0_CSR reset value.....                | 774  |
| 19.13.7        | CMU use restrictions.....                 | 774  |
| 19.13.8        | Loss-of-clock behavior.....               | 774  |
| 19.14          | PLL loss of lock interrupt.....           | 775  |
| 19.15          | Debug mode.....                           | 775  |

## Chapter 20

### Progressive Clock Switch (PCS)

|        |                                                       |     |
|--------|-------------------------------------------------------|-----|
| 20.1   | Introduction.....                                     | 777 |
| 20.2   | Block diagram.....                                    | 777 |
| 20.3   | Features.....                                         | 777 |
| 20.4   | Memory map and register definition.....               | 778 |
| 20.4.1 | PCS Switch Duration Register (PCS_SDUR).....          | 779 |
| 20.4.2 | PCS Divider Change Register for XOSC (PCS_DIVC1)..... | 779 |
| 20.4.3 | PCS Divider End Register for XOSC (PCS_DIVE1).....    | 780 |
| 20.4.4 | PCS Divider Start Register for XOSC (PCS_DIVS1).....  | 780 |
| 20.4.5 | PCS Divider Change Register for PLL1 (PCS_DIVC2)..... | 781 |
| 20.4.6 | PCS Divider End Register for PLL1 (PCS_DIVE2).....    | 782 |
| 20.4.7 | PCS Divider Start Register for PLL1 (PCS_DIVS2).....  | 782 |

| Section number | Title                                                 | Page |
|----------------|-------------------------------------------------------|------|
| 20.4.8         | PCS Divider Change Register for PLL0 (PCS_DIVC3)..... | 783  |
| 20.4.9         | PCS Divider End Register for PLL0 (PCS_DIVE3).....    | 783  |
| 20.4.10        | PCS Divider Start Register for PLL0 (PCS_DIVS3).....  | 784  |
| 20.5           | Functional description.....                           | 785  |
| 20.5.1         | Generic clock change requirements.....                | 786  |
| 20.5.2         | Configuration of PCS_SDUR.....                        | 786  |
| 20.5.3         | Configuration of PCS_DIVCn[RATE].....                 | 786  |
| 20.5.4         | Generic clock change properties.....                  | 787  |
| 20.5.5         | Clock ramp-down.....                                  | 787  |
| 20.5.6         | Clock ramp-up.....                                    | 788  |

## Chapter 21

### Dual PLL Digital Interface (PLLDIG)

|        |                                                                |     |
|--------|----------------------------------------------------------------|-----|
| 21.1   | Chip-specific PLLDIG information.....                          | 791 |
| 21.2   | Introduction.....                                              | 791 |
| 21.3   | Acronyms and Abbreviations.....                                | 791 |
| 21.4   | Block Diagram.....                                             | 792 |
| 21.5   | Features.....                                                  | 792 |
| 21.6   | Modes of operation.....                                        | 792 |
| 21.6.1 | Normal mode with reference, PLL0 or both PLLs enabled.....     | 792 |
| 21.7   | Memory map and register definition.....                        | 793 |
| 21.7.1 | PLLDIG PLL0 Control Register (PLLDIG_PLL0CR).....              | 794 |
| 21.7.2 | PLLDIG PLL0 Status Register (PLLDIG_PLL0SR).....               | 795 |
| 21.7.3 | PLLDIG PLL0 Divider Register (PLLDIG_PLL0DV).....              | 796 |
| 21.7.4 | PLLDIG PLL1 Control Register (PLLDIG_PLL1CR).....              | 798 |
| 21.7.5 | PLLDIG PLL1 Status Register (PLLDIG_PLL1SR).....               | 799 |
| 21.7.6 | PLLDIG PLL1 Divider Register (PLLDIG_PLL1DV).....              | 800 |
| 21.7.7 | PLLDIG PLL1 Frequency Modulation Register (PLLDIG_PLL1FM)..... | 801 |
| 21.7.8 | PLLDIG PLL1 Fractional Divide Register (PLLDIG_PLL1FD).....    | 802 |

| Section number | Title                           | Page |
|----------------|---------------------------------|------|
| 21.8           | Functional description.....     | 803  |
| 21.8.1         | Input clock frequency.....      | 803  |
| 21.8.2         | Clock configuration.....        | 804  |
| 21.8.3         | Loss of lock.....               | 806  |
| 21.8.4         | Frequency modulation.....       | 806  |
| 21.9           | Maximum lock time.....          | 809  |
| 21.10          | Initialization information..... | 809  |

## Chapter 22 Clock Monitor Unit (CMU)

|        |                                                                |     |
|--------|----------------------------------------------------------------|-----|
| 22.1   | Chip-specific CMU information.....                             | 811 |
| 22.2   | Introduction.....                                              | 811 |
| 22.2.1 | Main features.....                                             | 812 |
| 22.3   | Block diagram.....                                             | 812 |
| 22.4   | Signals.....                                                   | 812 |
| 22.5   | Register description and memory map.....                       | 813 |
| 22.5.1 | CMU Control Status Register (CMU_CSR).....                     | 814 |
| 22.5.2 | CMU Frequency Display Register (CMU_FDR).....                  | 815 |
| 22.5.3 | CMU High Frequency Reference Register CLKMN1 (CMU_HFREFR)..... | 816 |
| 22.5.4 | CMU Low Frequency Reference Register CLKMN1 (CMU_LFREFR).....  | 816 |
| 22.5.5 | CMU Interrupt Status Register (CMU_ISR).....                   | 817 |
| 22.5.6 | CMU Measurement Duration Register (CMU_MDR).....               | 818 |
| 22.6   | Functional description.....                                    | 819 |
| 22.6.1 | Frequency meter.....                                           | 819 |
| 22.6.2 | CLKMN0_RMT supervisor.....                                     | 819 |
| 22.6.3 | CLKMN1 supervisor.....                                         | 820 |

## Chapter 23 Embedded Memories

|      |                                                |     |
|------|------------------------------------------------|-----|
| 23.1 | Overview.....                                  | 823 |
| 23.2 | End-to-end Error Correction Code (e2eECC)..... | 823 |



## Chapter 24

### RAM Controller (PRAMC)

|        |                                                          |     |
|--------|----------------------------------------------------------|-----|
| 24.1   | Chip-specific PRAM information.....                      | 827 |
| 24.1.1 | PRAM optional read wait states.....                      | 827 |
| 24.1.2 | Reset effects on System RAM accesses.....                | 827 |
| 24.2   | Introduction.....                                        | 828 |
| 24.3   | SRAM controller memory map and register definition.....  | 830 |
| 24.3.1 | Platform RAM Configuration Register 1 (PRAMC_PRCR1)..... | 830 |
| 24.4   | Functional description.....                              | 832 |
| 24.4.1 | Read and Write operations.....                           | 832 |
| 24.4.2 | Writes.....                                              | 833 |
| 24.5   | Initialization/application information.....              | 835 |
| 24.6   | Reliability considerations.....                          | 835 |
| 24.6.1 | Hsiao ECC algorithm.....                                 | 835 |
| 24.6.2 | Transaction monitor.....                                 | 838 |

## Chapter 25

### Flash Memory Controller (PFLASH)

|        |                                                               |     |
|--------|---------------------------------------------------------------|-----|
| 25.1   | Chip-specific PFLASH information.....                         | 841 |
| 25.2   | Introduction.....                                             | 841 |
| 25.3   | Features.....                                                 | 842 |
| 25.4   | Block diagrams.....                                           | 842 |
| 25.5   | Flash memory controller memory map.....                       | 843 |
| 25.5.1 | Platform Flash Configuration Register 1 (PFLASH_PFCR1).....   | 845 |
| 25.5.2 | Platform Flash Configuration Register 2 (PFLASH_PFCR2).....   | 848 |
| 25.5.3 | Platform Flash Configuration Register 3 (PFLASH_PFCR3).....   | 851 |
| 25.5.4 | Platform Flash Access Protection Register (PFLASH_PFAPR)..... | 853 |
| 25.6   | Functional description.....                                   | 856 |
| 25.6.1 | Basic interface protocol.....                                 | 856 |

| Section number | Title                                         | Page |
|----------------|-----------------------------------------------|------|
| 25.6.2         | Access protections.....                       | 856  |
| 25.6.3         | Access pipelining.....                        | 857  |
| 25.6.4         | Line read buffers and prefetch operation..... | 858  |
| 25.6.5         | Instruction/Data prefetch triggering.....     | 859  |
| 25.6.6         | Per-Master prefetch triggering.....           | 859  |
| 25.6.7         | Buffer allocation.....                        | 859  |
| 25.6.8         | Reliability considerations.....               | 860  |
| 25.6.9         | ECC on data flash accesses.....               | 862  |
| 25.6.10        | Array integrity considerations.....           | 862  |

## Chapter 26 Embedded Flash Memory (C55FMC)

|        |                                                                            |     |
|--------|----------------------------------------------------------------------------|-----|
| 26.1   | Chip-specific C55FMC information.....                                      | 865 |
| 26.1.1 | C55FMC register reset values.....                                          | 865 |
| 26.1.2 | Access to and configuration of blocks mapped to C55FMC_LOCK1 register..... | 865 |
| 26.1.3 | C55FMC_LOCKn register bit mapping.....                                     | 866 |
| 26.1.4 | C55FMC_SELn register bit mapping.....                                      | 869 |
| 26.1.5 | C55FMC_OPPn register bit mapping.....                                      | 873 |
| 26.2   | Introduction.....                                                          | 876 |
| 26.2.1 | Overview.....                                                              | 877 |
| 26.2.2 | Features.....                                                              | 877 |
| 26.2.3 | Modes of operation.....                                                    | 877 |
| 26.3   | UTest NVM block.....                                                       | 878 |
| 26.4   | Test mode disable seal.....                                                | 878 |
| 26.5   | Memory map and register definition.....                                    | 880 |
| 26.5.1 | Module Configuration Register (C55FMC_MCR).....                            | 882 |
| 26.5.2 | Extended Module Configuration Register (C55FMC_MCRE).....                  | 887 |
| 26.5.3 | Lock 0 register (C55FMC_LOCK0).....                                        | 890 |
| 26.5.4 | Lock 1 register (C55FMC_LOCK1).....                                        | 892 |
| 26.5.5 | Lock 2 register (C55FMC_LOCK2).....                                        | 892 |

| Section number | Title                                                       | Page |
|----------------|-------------------------------------------------------------|------|
| 26.5.6         | Lock 3 register (C55FMC_LOCK3).....                         | 893  |
| 26.5.7         | Select 0 register (C55FMC_SEL0).....                        | 894  |
| 26.5.8         | Select 1 register (C55FMC_SEL1).....                        | 895  |
| 26.5.9         | Select 2 register (C55FMC_SEL2).....                        | 896  |
| 26.5.10        | Select 3 register (C55FMC_SEL3).....                        | 897  |
| 26.5.11        | Address register (C55FMC_ADR).....                          | 898  |
| 26.5.12        | UTest 0 register (C55FMC_UT0).....                          | 900  |
| 26.5.13        | UMISR register (C55FMC_UMn).....                            | 903  |
| 26.5.14        | UMISR register (C55FMC_UM9).....                            | 904  |
| 26.5.15        | Over-Program Protection 0 register (C55FMC_OPP0).....       | 905  |
| 26.5.16        | Over-Program Protection 1 register (C55FMC_OPP1).....       | 906  |
| 26.5.17        | Over-Program Protection 2 register (C55FMC_OPP2).....       | 907  |
| 26.5.18        | Over-Program Protection 3 register (C55FMC_OPP3).....       | 907  |
| 26.5.19        | Test Mode Disable Password Check register (C55FMC_TMD)..... | 908  |
| 26.6           | Functional Description.....                                 | 909  |
| 26.6.1         | User Mode.....                                              | 910  |
| 26.6.2         | UTest mode.....                                             | 918  |
| 26.7           | Initialization information.....                             | 922  |

## Chapter 27 Analog Modules Overview

|      |                                         |     |
|------|-----------------------------------------|-----|
| 27.1 | Introduction .....                      | 925 |
| 27.2 | EQADC subsystem.....                    | 925 |
| 27.3 | SDADC connections to other modules..... | 927 |

## Chapter 28 Sigma Delta Analog-to-Digital Converter (SDADC) Digital Interface

|        |                                           |     |
|--------|-------------------------------------------|-----|
| 28.1   | Chip-specific SDADC information.....      | 929 |
| 28.1.1 | SDADC triggers and synchronous start..... | 929 |
| 28.1.2 | Connections among SDADCs and eTPUs.....   | 931 |
| 28.2   | Introduction.....                         | 931 |

| Section number | Title                                                | Page |
|----------------|------------------------------------------------------|------|
| 28.3           | Overview.....                                        | 931  |
| 28.4           | Features.....                                        | 932  |
| 28.5           | Modes of operation.....                              | 933  |
| 28.5.1         | Differential input mode.....                         | 933  |
| 28.5.2         | Single-ended input mode.....                         | 933  |
| 28.6           | External signal description.....                     | 934  |
| 28.7           | Memory map and register definition.....              | 934  |
| 28.7.1         | Module Configuration Register (SDADC_MCR).....       | 935  |
| 28.7.2         | Channel Selection Register (SDADC_CSR).....          | 938  |
| 28.7.3         | Reset Key Register (SDADC_RKR).....                  | 940  |
| 28.7.4         | Status Flag Register (SDADC_SFR).....                | 941  |
| 28.7.5         | Request Select and Enable Register (SDADC_RSER)..... | 943  |
| 28.7.6         | Output Settling Delay Register (SDADC_OSDR).....     | 945  |
| 28.7.7         | FIFO Control Register (SDADC_FCR).....               | 946  |
| 28.7.8         | Software Trigger Key Register (SDADC_STKR).....      | 948  |
| 28.7.9         | Converted Data Register (SDADC_CDR).....             | 948  |
| 28.7.10        | WDG Threshold Register (SDADC_WTHHLR).....           | 949  |
| 28.8           | Functional description.....                          | 949  |
| 28.8.1         | Differential input mode.....                         | 950  |
| 28.8.2         | Single-ended input mode.....                         | 950  |
| 28.8.3         | Low consumption mode.....                            | 950  |
| 28.8.4         | Analog input multiplexing and bias support.....      | 950  |
| 28.8.5         | Programmable gain and decimation rate.....           | 951  |
| 28.8.6         | High-pass filter support.....                        | 951  |
| 28.8.7         | Data conversion.....                                 | 951  |
| 28.8.8         | Hardware triggering.....                             | 953  |
| 28.8.9         | Interrupt/DMA request support.....                   | 954  |
| 28.8.10        | Gain calibration support.....                        | 954  |
| 28.8.11        | Offset calibration support.....                      | 955  |

| Section number | Title                           | Page |
|----------------|---------------------------------|------|
| 28.9           | Initialization information..... | 956  |
| 28.9.1         | Data conversion step.....       | 957  |

## Chapter 29

### Enhanced Queued Analog-to-Digital Converter (EQADC)

|        |                                                                   |     |
|--------|-------------------------------------------------------------------|-----|
| 29.1   | Chip-specific EQADC information.....                              | 959 |
| 29.1.1 | EQADC connections to other modules.....                           | 960 |
| 29.1.2 | EQADC Stop mode entry.....                                        | 960 |
| 29.1.3 | EQADC pre-gain control.....                                       | 960 |
| 29.1.4 | CFIFO and RFIFO depth.....                                        | 961 |
| 29.1.5 | EQADC pin pullup/pulldown control.....                            | 961 |
| 29.1.6 | EQADC pin-to-channel mapping.....                                 | 961 |
| 29.2   | Introduction.....                                                 | 963 |
| 29.2.1 | Module overview.....                                              | 963 |
| 29.2.2 | Block diagram.....                                                | 963 |
| 29.2.3 | Features.....                                                     | 966 |
| 29.3   | Modes of operation.....                                           | 967 |
| 29.3.1 | Normal Mode.....                                                  | 967 |
| 29.3.2 | Debug Mode.....                                                   | 967 |
| 29.3.3 | Stop Mode.....                                                    | 969 |
| 29.4   | External signal description.....                                  | 970 |
| 29.4.1 | Overview.....                                                     | 970 |
| 29.4.2 | Detailed signal descriptions.....                                 | 972 |
| 29.5   | Memory map and register definition.....                           | 977 |
| 29.5.1 | EQADC Module Configuration Register (EQADC_MCR).....              | 981 |
| 29.5.2 | EQADC Test Register (EQADC_TST).....                              | 982 |
| 29.5.3 | EQADC External Trigger Digital Filter Register (EQADC_ETDFR)..... | 983 |
| 29.5.4 | EQADC CFIFO Push Register (EQADC_CFPR <sub>n</sub> ).....         | 984 |
| 29.5.5 | EQADC Result FIFO Pop Register (EQADC_RFPR <sub>n</sub> ).....    | 985 |
| 29.5.6 | EQADC CFIFO Control Register 0 (EQADC_CFCR0).....                 | 986 |

| Section number | Title                                                                    | Page |
|----------------|--------------------------------------------------------------------------|------|
| 29.5.7         | EQADC CFIFO Control Register 1 (EQADC_CFCR1).....                        | 990  |
| 29.5.8         | EQADC CFIFO Control Register 2 (EQADC_CFCR2).....                        | 993  |
| 29.5.9         | EQADC Interrupt and DMA Control Register 0 (EQADC_IDCR0).....            | 996  |
| 29.5.10        | EQADC Interrupt and DMA Control Register 1 (EQADC_IDCR1).....            | 1000 |
| 29.5.11        | EQADC Interrupt and DMA Control Register 2 (EQADC_IDCR2).....            | 1005 |
| 29.5.12        | EQADC FIFO and Interrupt Status Register (EQADC_FISR <sub>n</sub> )..... | 1010 |
| 29.5.13        | EQADC CFIFO Transfer Counter Register 0 (EQADC_CFTCR0).....              | 1014 |
| 29.5.14        | EQADC CFIFO Transfer Counter Register 1 (EQADC_CFTCR1).....              | 1015 |
| 29.5.15        | EQADC CFIFO Transfer Counter Register 2 (EQADC_CFTCR2).....              | 1016 |
| 29.5.16        | EQADC CFIFO Status Snapshot Register 0 (EQADC_CFSSR0).....               | 1017 |
| 29.5.17        | EQADC CFIFO Status Snapshot Register 1 (EQADC_CFSSR1).....               | 1019 |
| 29.5.18        | EQADC CFIFO Status Register (EQADC_CFSR).....                            | 1021 |
| 29.5.19        | EQADC STAC Bus Client Configuration Register (EQADC_STACCCR).....        | 1022 |
| 29.5.20        | EQADC CFIFO0 Register (EQADC_CF0R <sub>n</sub> ).....                    | 1023 |
| 29.5.21        | EQADC CFIFO0 Extension Register (EQADC_CF0ER <sub>n</sub> ).....         | 1024 |
| 29.5.22        | EQADC CFIFO1 Register (EQADC_CF1R <sub>n</sub> ).....                    | 1024 |
| 29.5.23        | EQADC CFIFO2 Register (EQADC_CF2R <sub>n</sub> ).....                    | 1025 |
| 29.5.24        | EQADC CFIFO3 Register (EQADC_CF3R <sub>n</sub> ).....                    | 1026 |
| 29.5.25        | EQADC CFIFO4 Register (EQADC_CF4R <sub>n</sub> ).....                    | 1026 |
| 29.5.26        | EQADC CFIFO5 Register (EQADC_CF5R <sub>n</sub> ).....                    | 1027 |
| 29.5.27        | EQADC RFIFO0 Register (EQADC_RF0R <sub>n</sub> ).....                    | 1028 |
| 29.5.28        | EQADC RFIFO1 Register (EQADC_RF1R <sub>n</sub> ).....                    | 1028 |
| 29.5.29        | EQADC RFIFO2 Register (EQADC_RF2R <sub>n</sub> ).....                    | 1029 |
| 29.5.30        | EQADC RFIFO3 Register (EQADC_RF3R <sub>n</sub> ).....                    | 1030 |
| 29.5.31        | EQADC RFIFO4 Register (EQADC_RF4R <sub>n</sub> ).....                    | 1030 |
| 29.5.32        | EQADC RFIFO5 Register (EQADC_RF5R <sub>n</sub> ).....                    | 1031 |
| 29.5.33        | On-chip ADC registers.....                                               | 1031 |
| 29.6           | Functional description.....                                              | 1054 |
| 29.6.1         | Overview.....                                                            | 1054 |

| Section number | Title                                              | Page |
|----------------|----------------------------------------------------|------|
| 29.6.2         | Data flow in EQADC.....                            | 1055 |
| 29.6.3         | Command/result queues.....                         | 1068 |
| 29.6.4         | EQADC command FIFOs.....                           | 1069 |
| 29.6.5         | EQADC result FIFOs.....                            | 1096 |
| 29.6.6         | On-chip ADC configuration and control.....         | 1100 |
| 29.6.7         | Internal/external multiplexing.....                | 1113 |
| 29.6.8         | EQADC DMA/interrupt request.....                   | 1117 |
| 29.6.9         | EQADC Parallel Side Interface (PSI) sub-block..... | 1120 |
| 29.6.10        | Analog sub-block.....                              | 1124 |
| 29.7           | Initialization/application information.....        | 1128 |
| 29.7.1         | Multiple queues control setup example.....         | 1128 |
| 29.7.2         | EQADC/DMAC interface.....                          | 1133 |
| 29.7.3         | Sending immediate command setup example.....       | 1135 |
| 29.7.4         | Modifying queues.....                              | 1136 |
| 29.7.5         | CQueue and RQueues usage.....                      | 1137 |
| 29.7.6         | ADC Result Calibration.....                        | 1139 |
| 29.7.7         | EQADC versus QADC.....                             | 1142 |

## Chapter 30 Temperature Sensor (TSENS)

|        |                                                                  |      |
|--------|------------------------------------------------------------------|------|
| 30.1   | Chip-specific Temperature Sensor information.....                | 1147 |
| 30.1.1 | Temperature Sensor programming control.....                      | 1147 |
| 30.1.2 | Accessing TSC parameters.....                                    | 1147 |
| 30.1.3 | Temperature name alignment.....                                  | 1147 |
| 30.2   | Introduction.....                                                | 1148 |
| 30.3   | Functional Description.....                                      | 1148 |
| 30.3.1 | Temperature threshold detection (digital output generation)..... | 1148 |
| 30.3.2 | Linear temperature sensor (analog output generation).....        | 1149 |
| 30.4   | Temperature formula.....                                         | 1150 |
| 30.5   | Calculating device temperature.....                              | 1151 |

## Chapter 31

### System Timer Module (STM)

|        |                                                             |      |
|--------|-------------------------------------------------------------|------|
| 31.1   | Chip-specific STM information.....                          | 1153 |
| 31.2   | Introduction.....                                           | 1153 |
| 31.2.1 | Overview.....                                               | 1153 |
| 31.2.2 | Features.....                                               | 1153 |
| 31.2.3 | Modes of operation.....                                     | 1154 |
| 31.3   | External signal description.....                            | 1154 |
| 31.4   | Memory map and registers.....                               | 1154 |
| 31.4.1 | STM Control Register (STM_CR).....                          | 1155 |
| 31.4.2 | STM Count Register (STM_CNT).....                           | 1156 |
| 31.4.3 | STM Channel Control Register (STM_CCR <sub>n</sub> ).....   | 1156 |
| 31.4.4 | STM Channel Interrupt Register (STM_CIR <sub>n</sub> )..... | 1157 |
| 31.4.5 | STM Channel Compare Register (STM_CMP <sub>n</sub> ).....   | 1157 |
| 31.5   | Functional description.....                                 | 1158 |

## Chapter 32

### Software Watchdog Timer (SWT)

|        |                                                                                         |      |
|--------|-----------------------------------------------------------------------------------------|------|
| 32.1   | Chip-specific SWT information.....                                                      | 1159 |
| 32.1.1 | SWT register reset values.....                                                          | 1159 |
| 32.1.2 | SWT time-out events and invalid accesses: configuring reset and interrupt requests..... | 1159 |
| 32.2   | Introduction.....                                                                       | 1160 |
| 32.2.1 | Overview.....                                                                           | 1160 |
| 32.2.2 | Features.....                                                                           | 1161 |
| 32.2.3 | Modes of operation.....                                                                 | 1161 |
| 32.3   | External signal description.....                                                        | 1162 |
| 32.4   | Memory Map and Registers.....                                                           | 1162 |
| 32.4.1 | SWT Control Register (SWT_CR).....                                                      | 1163 |
| 32.4.2 | SWT Interrupt Register (SWT_IR).....                                                    | 1166 |
| 32.4.3 | SWT Time-out Register (SWT_TO).....                                                     | 1167 |



| Section number | Title                                     | Page |
|----------------|-------------------------------------------|------|
| 32.4.4         | SWT Window Register (SWT_WN).....         | 1167 |
| 32.4.5         | SWT Service Register (SWT_SR).....        | 1168 |
| 32.4.6         | SWT Counter Output Register (SWT_CO)..... | 1168 |
| 32.4.7         | SWT Service Key Register (SWT_SK).....    | 1169 |
| 32.5           | Functional description.....               | 1169 |
| 32.5.1         | Introduction.....                         | 1169 |
| 32.5.2         | Configuration locking.....                | 1170 |
| 32.5.3         | Unlock sequence.....                      | 1171 |
| 32.5.4         | Servicing operations.....                 | 1171 |
| 32.5.5         | Time-out.....                             | 1172 |
| 32.5.6         | Initialization.....                       | 1173 |

## Chapter 33

### Periodic Interrupt Timer (PIT-RTI)

|         |                                                      |      |
|---------|------------------------------------------------------|------|
| 33.1    | Chip-specific PIT information.....                   | 1175 |
| 33.2    | Introduction.....                                    | 1175 |
| 33.2.1  | Block diagram.....                                   | 1175 |
| 33.2.2  | Features.....                                        | 1176 |
| 33.3    | Signal description.....                              | 1177 |
| 33.4    | Memory map/register description.....                 | 1177 |
| 33.4.1  | PIT Module Control Register (PIT_MCR).....           | 1178 |
| 33.4.2  | PIT Upper Lifetime Timer Register (PIT_LTMR64H)..... | 1179 |
| 33.4.3  | PIT Lower Lifetime Timer Register (PIT_LTMR64L)..... | 1180 |
| 33.4.4  | Timer Load Value Register (PIT_RTI_LDVAL).....       | 1180 |
| 33.4.5  | Current Timer Value Register (PIT_RTI_CVAL).....     | 1181 |
| 33.4.6  | Timer Control Register (PIT_RTI_TCTRL).....          | 1181 |
| 33.4.7  | Timer Flag Register (PIT_RTI_TFLG).....              | 1182 |
| 33.4.8  | Timer Load Value Register (PIT_LDVALn).....          | 1183 |
| 33.4.9  | Current Timer Value Register (PIT_CVALn).....        | 1183 |
| 33.4.10 | Timer Control Register (PIT_TCTRLn).....             | 1184 |

| Section number | Title                                             | Page |
|----------------|---------------------------------------------------|------|
| 33.4.11        | Timer Flag Register (PIT_TFLGn).....              | 1185 |
| 33.5           | Functional description.....                       | 1185 |
| 33.5.1         | General operation.....                            | 1185 |
| 33.5.2         | Interrupts.....                                   | 1187 |
| 33.5.3         | Chained timers.....                               | 1188 |
| 33.6           | Initialization and application information.....   | 1188 |
| 33.7           | Example configuration for chained timers.....     | 1189 |
| 33.8           | Example configuration for the lifetime timer..... | 1190 |

## Chapter 34

### Enhanced Modular IO Subsystem (eMIOS)

|        |                                                         |      |
|--------|---------------------------------------------------------|------|
| 34.1   | Chip-specific eMIOS information.....                    | 1191 |
| 34.1.1 | eMIOS channels and counter buses.....                   | 1191 |
| 34.1.2 | eMIOS unimplemented channels.....                       | 1191 |
| 34.1.3 | eMIOS interaction with other modules.....               | 1192 |
| 34.2   | Introduction.....                                       | 1194 |
| 34.2.1 | Overview.....                                           | 1194 |
| 34.2.2 | Features.....                                           | 1195 |
| 34.2.3 | Modes of operation.....                                 | 1196 |
| 34.3   | Memory map / register definition.....                   | 1197 |
| 34.3.1 | Memory map.....                                         | 1197 |
| 34.4   | Control Memory Map and Registers.....                   | 1198 |
| 34.4.1 | eMIOS Module Configuration Register (eMIOS_MCR).....    | 1199 |
| 34.4.2 | eMIOS Global FLAG Register (eMIOS_GFLAG).....           | 1201 |
| 34.4.3 | eMIOS Output Update Disable register (eMIOS_OUDIS)..... | 1201 |
| 34.5   | Unified Channels Memory Map and Registers.....          | 1202 |
| 34.5.1 | eMIOS UC A register n (eMIOS_An).....                   | 1205 |
| 34.5.2 | eMIOS UC B register n (eMIOS_Bn).....                   | 1206 |
| 34.5.3 | eMIOS CNT register n (eMIOS_CNTn).....                  | 1207 |
| 34.5.4 | eMIOS UC Control register n (eMIOS_Cn).....             | 1208 |

| Section number | Title                                                          | Page |
|----------------|----------------------------------------------------------------|------|
| 34.5.5         | eMIOS UC Status register n (eMIOS_ <i>Sn</i> ).....            | 1214 |
| 34.5.6         | eMIOS Alternate Address register n (eMIOS_ <i>ALTAn</i> )..... | 1216 |
| 34.5.7         | eMIOS UC A register n (eMIOS_ <i>An</i> ).....                 | 1217 |
| 34.5.8         | eMIOS UC B register n (eMIOS_ <i>Bn</i> ).....                 | 1217 |
| 34.5.9         | eMIOS CNT register n (eMIOS_ <i>CNTn</i> ).....                | 1218 |
| 34.5.10        | eMIOS UC Control register n (eMIOS_ <i>Cn</i> ).....           | 1219 |
| 34.5.11        | eMIOS UC Status register n (eMIOS_ <i>Sn</i> ).....            | 1224 |
| 34.5.12        | eMIOS Alternate Address register n (eMIOS_ <i>ALTAn</i> )..... | 1226 |
| 34.6           | Functional Description.....                                    | 1226 |
| 34.6.1         | Unified Channel (UC).....                                      | 1228 |
| 34.6.2         | IP Bus Interface Unit (BIU).....                               | 1268 |
| 34.6.3         | STAC Bus Client submodule.....                                 | 1269 |
| 34.6.4         | Global Clock Prescaler Submodule (GCP).....                    | 1270 |
| 34.7           | Initialization/Application Information.....                    | 1270 |
| 34.7.1         | Considerations.....                                            | 1270 |
| 34.7.2         | Application Information.....                                   | 1271 |

## Chapter 35

### Enhanced Time Processing Unit (eTPU)

|        |                                                           |      |
|--------|-----------------------------------------------------------|------|
| 35.1   | Chip-specific eTPU information.....                       | 1275 |
| 35.1.1 | eTPU register field reset values.....                     | 1276 |
| 35.1.2 | Access to eTPU registers in chip-level Stop mode.....     | 1276 |
| 35.1.3 | STAC server IDs.....                                      | 1276 |
| 35.1.4 | eTPU inputs with Glitch Filters.....                      | 1277 |
| 35.1.5 | eMIOS and eTPU signal passthrough from pins via IGFs..... | 1277 |
| 35.1.6 | eTPU output disable channel groups.....                   | 1278 |
| 35.1.7 | Control of the eTPU Output Buffer Enable signal.....      | 1278 |
| 35.2   | Device-specific Information.....                          | 1278 |

| Section number | Title                                                                          | Page |
|----------------|--------------------------------------------------------------------------------|------|
| 35.3           | Introduction.....                                                              | 1279 |
| 35.3.1         | Overview.....                                                                  | 1280 |
| 35.3.2         | Features.....                                                                  | 1287 |
| 35.3.3         | Modes of operation.....                                                        | 1294 |
| 35.4           | Detailed signal descriptions.....                                              | 1296 |
| 35.4.1         | eTPU channel output signals.....                                               | 1296 |
| 35.4.2         | eTPU channel input signals.....                                                | 1296 |
| 35.4.3         | Time Base Clock Signal — TCRCLK.....                                           | 1297 |
| 35.4.4         | eTPU channel output disable signals.....                                       | 1297 |
| 35.5           | Memory map/register definition.....                                            | 1298 |
| 35.6           | Memory map.....                                                                | 1298 |
| 35.7           | System Configuration memory map and registers.....                             | 1303 |
| 35.7.1         | eTPU Module Configuration Register (eTPU_SCR_MCR).....                         | 1304 |
| 35.7.2         | eTPU Coherent Dual-Parameter Controller Register (eTPU_SCR_CDCR).....          | 1308 |
| 35.7.3         | eTPU MISC Compare Register (eTPU_SCR_MISCCMPR).....                            | 1310 |
| 35.7.4         | eTPU SCM Off-range Data Register (eTPU_SCR_SCMOFFDATAR).....                   | 1310 |
| 35.7.5         | eTPU Engine 1 Configuration Register (eTPU_SCR_ECR_ENG1).....                  | 1311 |
| 35.7.6         | eTPU Engine 2 Configuration Register (eTPU_SCR_ECR_ENG2).....                  | 1315 |
| 35.8           | Timebase memory map and registers.....                                         | 1319 |
| 35.8.1         | eTPU Engine 1 Time Base Configuration Register (eTPU_TBR_TBCR_ENG1).....       | 1320 |
| 35.8.2         | eTPU Engine 1 Time Base 1 (TCR1) Visibility Register (eTPU_TBR_TB1R_ENG1)..... | 1324 |
| 35.8.3         | eTPU Engine 1 Time Base 2 (TCR2) Visibility Register (eTPU_TBR_TB2R_ENG1)..... | 1324 |
| 35.8.4         | eTPU Engine 1 STAC Configuration Register (eTPU_TBR_STAC_ENG1).....            | 1325 |
| 35.8.5         | eTPU Engine 2 Time Base Configuration Register (eTPU_TBR_TBCR_ENG2).....       | 1327 |
| 35.8.6         | eTPU Engine 2 Time Base 1 (TCR1) Visibility Register (eTPU_TBR_TB1R_ENG2)..... | 1331 |
| 35.8.7         | eTPU Engine 2 Time Base 2 (TCR2) Visibility Register (eTPU_TBR_TB2R_ENG2)..... | 1331 |
| 35.8.8         | eTPU Engine 2 STAC Configuration Register (eTPU_TBR_STAC_ENG2).....            | 1332 |
| 35.9           | Engine memory map and registers.....                                           | 1334 |
| 35.9.1         | eTPU Watchdog Timer Register for Engine 1 (eTPU_ER_WDTR_ENG1).....             | 1334 |

| Section number | Title                                                                                       | Page |
|----------------|---------------------------------------------------------------------------------------------|------|
| 35.9.2         | eTPU Idle Register for Engine 1 (eTPU_ER_IDLER_ENG1).....                                   | 1336 |
| 35.9.3         | eTPU Watchdog Timer Register for Engine 2 (eTPU_ER_WDTR_ENG2).....                          | 1337 |
| 35.9.4         | eTPU Idle Register for Engine 2 (eTPU_ER_IDLER_ENG2).....                                   | 1338 |
| 35.10          | Latency Monitor memory map and registers.....                                               | 1339 |
| 35.10.1        | eTPU Latency Monitor Configuration Register for Engine 1 (eTPU_LMR_LMnCR_ENG1).....         | 1340 |
| 35.10.2        | eTPU Latency Monitor Measure Register for Engine 1 (eTPU_LMR_LMnMR_ENG1).....               | 1342 |
| 35.10.3        | eTPU Latency Monitor Configuration Register for Engine 2 (eTPU_LMR_LMnCR_ENG2).....         | 1343 |
| 35.10.4        | eTPU Latency Monitor Measure Register for Engine 2 (eTPU_LMR_LMnMR_ENG2).....               | 1344 |
| 35.11          | Memory Error Support memory map and registers.....                                          | 1345 |
| 35.11.1        | eTPU Memory Error Control Register (eTPU_MESR_MECR).....                                    | 1346 |
| 35.11.2        | eTPU Data Error Injection Address Register (eTPU_MESR_DEIAR).....                           | 1349 |
| 35.11.3        | eTPU Data Error Injection Data Pattern Register (eTPU_MESR_DEIDPR).....                     | 1350 |
| 35.11.4        | eTPU Data Error Injection Parity Pattern Register (eTPU_MESR_DEIPPR).....                   | 1350 |
| 35.11.5        | eTPU Data Error Report Address Register (eTPU_MESR_DERAR).....                              | 1352 |
| 35.11.6        | eTPU Data Error Report Data Register (eTPU_MESR_DERDR).....                                 | 1353 |
| 35.11.7        | eTPU Data Error Report Syndrome Register (eTPU_MESR_DERSR).....                             | 1354 |
| 35.11.8        | eTPU Memory Error Status Register (eTPU_MESR_MESR).....                                     | 1356 |
| 35.11.9        | eTPU Code Error Injection Address Register (eTPU_MESR_CEIAR).....                           | 1359 |
| 35.11.10       | eTPU Code Error Injection Data Pattern Register (eTPU_MESR_CEIDPR).....                     | 1360 |
| 35.11.11       | eTPU Code Error Injection Parity Pattern Register (eTPU_MESR_CEIPPR).....                   | 1360 |
| 35.11.12       | eTPU Code Error Report Address Register (eTPU_MESR_CERAR).....                              | 1361 |
| 35.11.13       | eTPU Code Error Report Data Register (eTPU_MESR_CERDR).....                                 | 1362 |
| 35.11.14       | eTPU Code Error Report Syndrome Register (eTPU_MESR_CERSR).....                             | 1363 |
| 35.11.15       | eTPU Code Error Fix Register (eTPU_MESR_CEFR).....                                          | 1364 |
| 35.12          | Global Channel memory map and registers.....                                                | 1365 |
| 35.12.1        | eTPU Channel Interrupt Status Register for Engine 1 (eTPU_GCR_CISR_ENG1).....               | 1366 |
| 35.12.2        | eTPU Channel Interrupt Status Register for Engine 2 (eTPU_GCR_CISR_ENG2).....               | 1366 |
| 35.12.3        | eTPU Channel Data Transfer Request Status Register for Engine 1 (eTPU_GCR_CDTRSR_ENG1)..... | 1367 |
| 35.12.4        | eTPU Channel Data Transfer Request Status Register for Engine 2 (eTPU_GCR_CDTRSR_ENG2)..... | 1368 |

| Section number | Title                                                                                                    | Page |
|----------------|----------------------------------------------------------------------------------------------------------|------|
| 35.12.5        | eTPU Channel Interrupt Overflow Status Register for Engine 1 (eTPU_GCR_CIOSR_ENG1).....                  | 1368 |
| 35.12.6        | eTPU Channel Interrupt Overflow Status Register for Engine 2 (eTPU_GCR_CIOSR_ENG2).....                  | 1369 |
| 35.12.7        | eTPU Channel Data Transfer Request Overflow Status Register for Engine 1<br>(eTPU_GCR_CDTROSR_ENG1)..... | 1370 |
| 35.12.8        | eTPU Channel Data Transfer Request Overflow Status Register for Engine 2<br>(eTPU_GCR_CDTROSR_ENG2)..... | 1370 |
| 35.12.9        | eTPU Channel Interrupt Enable Register for Engine 1 (eTPU_GCR_CIER_ENG1).....                            | 1371 |
| 35.12.10       | eTPU Channel Interrupt Enable Register for Engine 2 (eTPU_GCR_CIER_ENG2).....                            | 1371 |
| 35.12.11       | eTPU Channel Data Transfer Request Enable Register for Engine 1 (eTPU_GCR_CDTRER_ENG1)....               | 1372 |
| 35.12.12       | eTPU Channel Data Transfer Request Enable Register for Engine 2 (eTPU_GCR_CDTRER_ENG2)....               | 1372 |
| 35.12.13       | eTPU Watchdog Status Register for Engine 1 (eTPU_GCR_WDSR_ENG1).....                                     | 1373 |
| 35.12.14       | eTPU Watchdog Status Register for Engine 2 (eTPU_GCR_WDSR_ENG2).....                                     | 1374 |
| 35.12.15       | eTPU Channel Pending Service Status Register for Engine 1 (eTPU_GCR_CPSSR_ENG1).....                     | 1374 |
| 35.12.16       | eTPU Channel Pending Service Status Register for Engine 2 (eTPU_GCR_CPSSR_ENG2).....                     | 1375 |
| 35.12.17       | eTPU Channel Service Status Register for Engine 1 (eTPU_GCR_CSSR_ENG1).....                              | 1376 |
| 35.12.18       | eTPU Channel Service Status Register for Engine 2 (eTPU_GCR_CSSR_ENG2).....                              | 1376 |
| 35.13          | Channel Configuration Control memory map and registers.....                                              | 1377 |
| 35.13.1        | eTPU Channel n Configuration Register for Engine 1 (eTPU_CCCR_CnCR_ENG1).....                            | 1387 |
| 35.13.2        | eTPU Channel n Status Control Register for Engine 1 (eTPU_CCCR_CnSCR_ENG1).....                          | 1390 |
| 35.13.3        | eTPU Channel n Host Service Request Register for Engine 1 (eTPU_CCCR_CnHSRR_ENG1).....                   | 1392 |
| 35.13.4        | eTPU Channel n Configuration Register for Engine 2 (eTPU_CCCR_CnCR_ENG2).....                            | 1393 |
| 35.13.5        | eTPU Channel n Status Control Register for Engine 2 (eTPU_CCCR_CnSCR_ENG2).....                          | 1396 |
| 35.13.6        | eTPU Channel n Host Service Request Register for Engine 2 (eTPU_CCCR_CnHSRR_ENG2).....                   | 1398 |
| 35.14          | Shared Subsystem Integration memory map and registers.....                                               | 1399 |
| 35.14.1        | eTPU Shared Subsystem n Integration Register (eTPU_SSIR_ETPUSSSIRn).....                                 | 1400 |

| Section number | Title                                       | Page |
|----------------|---------------------------------------------|------|
| 35.15          | Functional description.....                 | 1400 |
| 35.15.1        | Functions and threads.....                  | 1401 |
| 35.15.2        | Host interface.....                         | 1416 |
| 35.15.3        | Scheduler.....                              | 1425 |
| 35.15.4        | Parameter sharing and coherency.....        | 1433 |
| 35.15.5        | Enhanced Channels.....                      | 1439 |
| 35.15.6        | Time bases.....                             | 1493 |
| 35.15.7        | EAC - eTPU Angle Counter.....               | 1502 |
| 35.15.8        | Microengine.....                            | 1525 |
| 35.15.9        | Microinstruction Set.....                   | 1544 |
| 35.15.10       | Shared Subsystems support.....              | 1578 |
| 35.15.11       | Development support features.....           | 1582 |
| 35.15.12       | SCM Emulation.....                          | 1590 |
| 35.15.13       | Safety features.....                        | 1590 |
| 35.15.14       | Performance monitoring features.....        | 1598 |
| 35.16          | Initialization/application information..... | 1602 |
| 35.16.1        | Configuration sequence.....                 | 1602 |
| 35.16.2        | Reset options.....                          | 1604 |
| 35.16.3        | Multiple parameter coherency methods.....   | 1604 |
| 35.16.4        | Programming Hints and Caveats.....          | 1605 |
| 35.16.5        | Estimating Worst Case Latency.....          | 1607 |
| 35.16.6        | Endianness.....                             | 1625 |
| 35.16.7        | Memory Error Service Procedures.....        | 1626 |
| 35.17          | Appendices.....                             | 1629 |
| 35.17.1        | Microcycle and I/O timing.....              | 1629 |
| 35.17.2        | Channel Mode Summary.....                   | 1634 |
| 35.17.3        | MISC Algorithm.....                         | 1637 |

| Section number                   | Title                                                        | Page |
|----------------------------------|--------------------------------------------------------------|------|
| <b>Chapter 36</b>                |                                                              |      |
| <b>Input Glitch Filter (IGF)</b> |                                                              |      |
| 36.1                             | Chip-specific IGF information.....                           | 1639 |
| 36.1.1                           | IGF channels and internal prescalers.....                    | 1639 |
| 36.1.2                           | IGF internal prescaler selection.....                        | 1639 |
| 36.1.3                           | IGF interaction with other modules.....                      | 1640 |
| 36.1.4                           | STOP mode system requests for IGF_0 and IGF_1.....           | 1641 |
| 36.2                             | Introduction.....                                            | 1641 |
| 36.3                             | Block Diagram.....                                           | 1642 |
| 36.4                             | Input Glitch Filter overview.....                            | 1642 |
| 36.4.1                           | Interaction with other peripherals.....                      | 1643 |
| 36.4.2                           | Mode of operation.....                                       | 1643 |
| 36.5                             | Memory map.....                                              | 1645 |
| 36.5.1                           | Module Configuration Register (IGF_MCR <sub>n</sub> ).....   | 1650 |
| 36.5.2                           | Module Status Register (IGF_MSR <sub>n</sub> ).....          | 1653 |
| 36.5.3                           | Prescaler Register (IGF_PRESR <sub>n</sub> ).....            | 1654 |
| 36.5.4                           | Rising edge Threshold Register (IGF_RTHR <sub>n</sub> )..... | 1655 |
| 36.5.5                           | Falling edge Threshold Register (IGF_FTHR).....              | 1655 |
| 36.6                             | Functional description.....                                  | 1655 |
| 36.6.1                           | Input double sampling.....                                   | 1656 |
| 36.6.2                           | Filter types.....                                            | 1656 |
| 36.6.3                           | Application.....                                             | 1660 |
| 36.6.4                           | Programming guidelines.....                                  | 1663 |
| 36.6.5                           | On-the-fly filter reconfiguration.....                       | 1663 |
| 36.7                             | Power saving mode.....                                       | 1664 |
| 36.7.1                           | STOP mode.....                                               | 1664 |



| Section number       | Title                                                 | Page |
|----------------------|-------------------------------------------------------|------|
| <b>Chapter 37</b>    |                                                       |      |
| <b>FlexCAN (CAN)</b> |                                                       |      |
| 37.1                 | Chip-specific FlexCAN information.....                | 1667 |
| 37.1.1               | FlexCAN RAM.....                                      | 1667 |
| 37.1.2               | FlexCAN reset.....                                    | 1667 |
| 37.1.3               | CAN_MCR[SUPV] and access to FlexCAN registers.....    | 1667 |
| 37.2                 | Introduction.....                                     | 1667 |
| 37.2.1               | Overview.....                                         | 1668 |
| 37.2.2               | FlexCAN module features.....                          | 1669 |
| 37.2.3               | Modes of operation.....                               | 1671 |
| 37.3                 | FlexCAN signal descriptions.....                      | 1672 |
| 37.3.1               | CAN Rx .....                                          | 1672 |
| 37.3.2               | CAN Tx .....                                          | 1673 |
| 37.4                 | Memory map/register definition.....                   | 1673 |
| 37.4.1               | FlexCAN memory mapping.....                           | 1673 |
| 37.4.2               | Module Configuration Register (CAN_MCR).....          | 1679 |
| 37.4.3               | Control 1 register (CAN_CTRL1).....                   | 1683 |
| 37.4.4               | Free Running Timer (CAN_TIMER).....                   | 1686 |
| 37.4.5               | Rx Mailboxes Global Mask Register (CAN_RXMGMASK)..... | 1687 |
| 37.4.6               | Rx 14 Mask register (CAN_RX14MASK).....               | 1688 |
| 37.4.7               | Rx 15 Mask register (CAN_RX15MASK).....               | 1689 |
| 37.4.8               | Error Counter (CAN_ECR).....                          | 1690 |
| 37.4.9               | Error and Status 1 register (CAN_ESR1).....           | 1691 |
| 37.4.10              | Interrupt Masks 2 register (CAN_IMASK2).....          | 1695 |
| 37.4.11              | Interrupt Masks 1 register (CAN_IMASK1).....          | 1696 |
| 37.4.12              | Interrupt Flags 2 register (CAN_IFLAG2).....          | 1696 |
| 37.4.13              | Interrupt Flags 1 register (CAN_IFLAG1).....          | 1697 |
| 37.4.14              | Control 2 register (CAN_CTRL2).....                   | 1699 |
| 37.4.15              | Error and Status 2 register (CAN_ESR2).....           | 1703 |

| Section number | Title                                                       | Page |
|----------------|-------------------------------------------------------------|------|
| 37.4.16        | CRC Register (CAN_CRCCR).....                               | 1704 |
| 37.4.17        | Rx FIFO Global Mask register (CAN_RXFGMASK).....            | 1705 |
| 37.4.18        | Rx FIFO Information Register (CAN_RXFIR).....               | 1706 |
| 37.4.19        | Rx Individual Mask Registers (CAN_RXIMR <sub>n</sub> )..... | 1706 |
| 37.4.20        | Memory Error Control Register (CAN_MECR).....               | 1707 |
| 37.4.21        | Error Injection Address Register (CAN_ERRIAR).....          | 1709 |
| 37.4.22        | Error Injection Data Pattern Register (CAN_ERRIDPR).....    | 1711 |
| 37.4.23        | Error Injection Parity Pattern Register (CAN_ERRIPPR).....  | 1711 |
| 37.4.24        | Error Report Address Register (CAN_RERRAR).....             | 1712 |
| 37.4.25        | Error Report Data Register (CAN_RERRDR).....                | 1713 |
| 37.4.26        | Error Report Syndrome Register (CAN_RERRSYNR).....          | 1714 |
| 37.4.27        | Error Status Register (CAN_ERRSR).....                      | 1716 |
| 37.4.28        | Message buffer structure.....                               | 1717 |
| 37.4.29        | Rx FIFO structure.....                                      | 1723 |
| 37.5           | Functional description.....                                 | 1725 |
| 37.5.1         | Transmit process.....                                       | 1726 |
| 37.5.2         | Arbitration process.....                                    | 1726 |
| 37.5.3         | Receive process.....                                        | 1730 |
| 37.5.4         | Matching process.....                                       | 1732 |
| 37.5.5         | Move process.....                                           | 1737 |
| 37.5.6         | Data coherence.....                                         | 1738 |
| 37.5.7         | Rx FIFO.....                                                | 1742 |
| 37.5.8         | CAN protocol related features.....                          | 1743 |
| 37.5.9         | Clock domains and restrictions.....                         | 1750 |
| 37.5.10        | Modes of operation details.....                             | 1751 |
| 37.5.11        | Interrupts.....                                             | 1754 |
| 37.5.12        | Bus interface.....                                          | 1755 |

| Section number | Title                                          | Page |
|----------------|------------------------------------------------|------|
| 37.5.13        | Detection and Correction of Memory Errors..... | 1756 |
| 37.6           | Initialization/application information.....    | 1760 |
| 37.6.1         | FlexCAN initialization sequence.....           | 1760 |

## Chapter 38 Modular CAN (M\_CAN)

|         |                                                             |      |
|---------|-------------------------------------------------------------|------|
| 38.1    | Chip-specific M_CAN information.....                        | 1763 |
| 38.1.1  | M_CAN Message RAM allocation.....                           | 1763 |
| 38.1.2  | Introduction.....                                           | 1763 |
| 38.1.3  | Functional Description.....                                 | 1764 |
| 38.1.4  | External Signals.....                                       | 1766 |
| 38.2    | Overview.....                                               | 1767 |
| 38.2.1  | Features.....                                               | 1767 |
| 38.2.2  | Block Diagram.....                                          | 1768 |
| 38.2.3  | Dual Clock Sources.....                                     | 1770 |
| 38.2.4  | Dual Interrupt Lines.....                                   | 1770 |
| 38.3    | Memory Map and Register Description.....                    | 1771 |
| 38.3.1  | Core Release Register (M_CAN_CREL).....                     | 1773 |
| 38.3.2  | Endian Register (M_CAN_ENDN).....                           | 1774 |
| 38.3.3  | Data Bit Timing and Prescaler Register (M_CAN_DBTP).....    | 1775 |
| 38.3.4  | Test Register (M_CAN_TEST).....                             | 1776 |
| 38.3.5  | RAM Watchdog Register (M_CAN_RWD).....                      | 1778 |
| 38.3.6  | CC Control Register (M_CAN_CCCR).....                       | 1779 |
| 38.3.7  | Nominal Bit Timing and Prescaler Register (M_CAN_NBTP)..... | 1781 |
| 38.3.8  | Timestamp Counter Configuration Register (M_CAN_TSCC).....  | 1783 |
| 38.3.9  | Timestamp Counter Value Register (M_CAN_TSCV).....          | 1784 |
| 38.3.10 | Timeout Counter Configuration Register (M_CAN_TOCC).....    | 1784 |
| 38.3.11 | Timeout Counter Value Register (M_CAN_TOCV).....            | 1785 |
| 38.3.12 | Error Counter Register (M_CAN_ECR).....                     | 1786 |
| 38.3.13 | Protocol Status Register (M_CAN_PSR).....                   | 1787 |

| Section number | Title                                                                   | Page |
|----------------|-------------------------------------------------------------------------|------|
| 38.3.14        | Transmitter Delay Compensation Register (M_CAN_TDCR).....               | 1790 |
| 38.3.15        | Interrupt Register (M_CAN_IR).....                                      | 1791 |
| 38.3.16        | Interrupt Enable Register (M_CAN_IE).....                               | 1794 |
| 38.3.17        | Interrupt Line Select Register (M_CAN_ILS).....                         | 1797 |
| 38.3.18        | Interrupt Line Enable Register (M_CAN_ILE).....                         | 1801 |
| 38.3.19        | Global Filter Configuration Register (M_CAN_GFC).....                   | 1802 |
| 38.3.20        | Standard ID Filter Configuration Register (M_CAN_SIDFC).....            | 1803 |
| 38.3.21        | Extended ID Filter Configuration Register (M_CAN_XIDFC).....            | 1804 |
| 38.3.22        | Extended ID and Mask Register (M_CAN_XIDAM).....                        | 1804 |
| 38.3.23        | High Priority Message Status Register (M_CAN_HPMS).....                 | 1805 |
| 38.3.24        | New Data 1 Register (M_CAN_NDAT1).....                                  | 1806 |
| 38.3.25        | New Data 2 Register (M_CAN_NDAT2).....                                  | 1806 |
| 38.3.26        | Rx FIFO 0 Configuration Register (M_CAN_RXF0C).....                     | 1807 |
| 38.3.27        | Rx FIFO 0 Status Register (M_CAN_RXF0S).....                            | 1808 |
| 38.3.28        | Rx FIFO 0 Acknowledge Register (M_CAN_RXF0A).....                       | 1809 |
| 38.3.29        | Rx Buffer Configuration Register (M_CAN_RXBC).....                      | 1809 |
| 38.3.30        | Rx FIFO 1 Configuration Register (M_CAN_RXF1C).....                     | 1810 |
| 38.3.31        | Rx FIFO 1 Status Register (M_CAN_RXF1S).....                            | 1811 |
| 38.3.32        | Rx FIFO 1 Acknowledge Register (M_CAN_RXF1A).....                       | 1812 |
| 38.3.33        | Rx Buffer / FIFO Element Size Configuration Register (M_CAN_RXESC)..... | 1813 |
| 38.3.34        | Tx Buffer Configuration Register (M_CAN_TXBC).....                      | 1815 |
| 38.3.35        | Tx FIFO/Queue Status Register (M_CAN_TXFQS).....                        | 1816 |
| 38.3.36        | Tx Buffer Element Size Configuration (M_CAN_TXESC).....                 | 1817 |
| 38.3.37        | Tx Buffer Request Pending Register (M_CAN_TXBRP).....                   | 1818 |
| 38.3.38        | Tx Buffer Add Request Register (M_CAN_TXBAR).....                       | 1819 |
| 38.3.39        | Tx Buffer Cancellation Request Register (M_CAN_TXBCR).....              | 1819 |
| 38.3.40        | Tx Buffer Transmission Occurred Register (M_CAN_TXBTO).....             | 1820 |
| 38.3.41        | Tx Buffer Cancellation Finished Register (M_CAN_TXBCF).....             | 1820 |
| 38.3.42        | Tx Buffer Transmission Interrupt Enable Register (M_CAN_TXBTIE).....    | 1821 |

| Section number | Title                                                                         | Page |
|----------------|-------------------------------------------------------------------------------|------|
| 38.3.43        | Tx Buffer Cancellation Finished Interrupt Enable Register (M_CAN_TXBCIE)..... | 1821 |
| 38.3.44        | Tx Event FIFO Configuration Register (M_CAN_TXEFC).....                       | 1822 |
| 38.3.45        | Tx Event FIFO Status Register (M_CAN_TXEFS).....                              | 1823 |
| 38.3.46        | Tx Event FIFO Acknowledge Register (M_CAN_TXEFA).....                         | 1824 |
| 38.4           | Message RAM.....                                                              | 1824 |
| 38.4.1         | Rx Buffer and FIFO Element.....                                               | 1825 |
| 38.4.2         | Tx Buffer Element.....                                                        | 1827 |
| 38.4.3         | Tx Event FIFO Element.....                                                    | 1829 |
| 38.4.4         | Standard Message ID Filter Element.....                                       | 1830 |
| 38.4.5         | Extended Message ID Filter Element.....                                       | 1832 |
| 38.5           | Functional Description.....                                                   | 1833 |
| 38.5.1         | Operating Modes.....                                                          | 1833 |
| 38.5.2         | Timestamp Generation.....                                                     | 1843 |
| 38.5.3         | Timeout Counter.....                                                          | 1843 |
| 38.5.4         | Rx Handling.....                                                              | 1844 |
| 38.5.5         | Tx Handling.....                                                              | 1855 |
| 38.5.6         | FIFO Acknowledge Handling.....                                                | 1861 |

## Chapter 39

### Deserial Serial Peripheral Interface (DSPI)

|        |                                                         |      |
|--------|---------------------------------------------------------|------|
| 39.1   | Chip-specific DSPI information.....                     | 1863 |
| 39.1.1 | DSPI and eMIOS direct connections.....                  | 1863 |
| 39.1.2 | DSPI connections with SIU, eTPU, and eMIOS modules..... | 1864 |
| 39.2   | Introduction.....                                       | 1865 |
| 39.2.1 | Block Diagram.....                                      | 1865 |
| 39.2.2 | Features.....                                           | 1866 |
| 39.2.3 | Interface configurations.....                           | 1869 |
| 39.2.4 | Modes of Operation.....                                 | 1870 |
| 39.3   | Module signal descriptions.....                         | 1871 |
| 39.3.1 | HT—Hardware Trigger.....                                | 1872 |

| Section number | Title                                                                                       | Page |
|----------------|---------------------------------------------------------------------------------------------|------|
| 39.3.2         | PCS0/SS—Peripheral Chip Select/Slave Select.....                                            | 1872 |
| 39.3.3         | PCS1—PCS3—Peripheral Chip Selects 1–3.....                                                  | 1872 |
| 39.3.4         | PCS4—Peripheral Chip Select 4.....                                                          | 1873 |
| 39.3.5         | PCS5/PCSS—Peripheral Chip Select 5/Peripheral Chip Select Strobe.....                       | 1873 |
| 39.3.6         | SCK—Serial Clock.....                                                                       | 1873 |
| 39.3.7         | SIN—Serial Input.....                                                                       | 1873 |
| 39.3.8         | SOUT—Serial Output.....                                                                     | 1874 |
| 39.4           | Memory Map/Register Definition.....                                                         | 1874 |
| 39.4.1         | Module Configuration Register (DSPI_MCR).....                                               | 1876 |
| 39.4.2         | Transfer Count Register (DSPI_TCR).....                                                     | 1879 |
| 39.4.3         | Clock and Transfer Attributes Register (In Master Mode) (DSPI_CTAR <sub>n</sub> ).....      | 1880 |
| 39.4.4         | Clock and Transfer Attributes Register (In Slave Mode) (DSPI_CTAR <sub>n</sub> _SLAVE)..... | 1885 |
| 39.4.5         | Status Register (DSPI_SR).....                                                              | 1887 |
| 39.4.6         | DMA/Interrupt Request Select and Enable Register (DSPI_RSER).....                           | 1890 |
| 39.4.7         | PUSH TX FIFO Register In Master Mode (DSPI_PUSHR).....                                      | 1892 |
| 39.4.8         | PUSH TX FIFO Register In Slave Mode (DSPI_PUSHR_SLAVE).....                                 | 1894 |
| 39.4.9         | POP RX FIFO Register (DSPI_POPR).....                                                       | 1895 |
| 39.4.10        | Transmit FIFO Registers (DSPI_TXFR <sub>n</sub> ).....                                      | 1895 |
| 39.4.11        | Receive FIFO Registers (DSPI_RXFR <sub>n</sub> ).....                                       | 1896 |
| 39.4.12        | DSI Configuration Register 0 (DSPI_DSICR0).....                                             | 1897 |
| 39.4.13        | DSI Serialization Data Register 0 (DSPI_SDR0).....                                          | 1899 |
| 39.4.14        | DSI Alternate Serialization Data Register 0 (DSPI_ASDR0).....                               | 1900 |
| 39.4.15        | DSI Transmit Comparison Register 0 (DSPI_COMPR0).....                                       | 1900 |
| 39.4.16        | DSI Deserialization Data Register 0 (DSPI_DDR0).....                                        | 1901 |
| 39.4.17        | DSI Configuration Register 1 (DSPI_DSICR1).....                                             | 1902 |
| 39.4.18        | DSI Serialization Source Select Register 0 (DSPI_SSR0).....                                 | 1903 |
| 39.4.19        | DSI Parallel Input Select Register 0 (DSPI_PISR0).....                                      | 1904 |
| 39.4.20        | DSI Parallel Input Select Register 1 (DSPI_PISR1).....                                      | 1905 |
| 39.4.21        | DSI Parallel Input Select Register 2 (DSPI_PISR2).....                                      | 1906 |

| Section number | Title                                                                 | Page |
|----------------|-----------------------------------------------------------------------|------|
| 39.4.22        | DSI Parallel Input Select Register 3 (DSPI_PISR3).....                | 1907 |
| 39.4.23        | DSI Deserialized Data Interrupt Mask Register 0 (DSPI_DIMR0).....     | 1908 |
| 39.4.24        | DSI Deserialized Data Polarity Interrupt Register 0 (DSPI_DPIR0)..... | 1908 |
| 39.5           | Functional description.....                                           | 1909 |
| 39.5.1         | Start and Stop of module transfers.....                               | 1910 |
| 39.5.2         | Serial Peripheral Interface DSPI configuration.....                   | 1911 |
| 39.5.3         | Deserial Serial Interface (DSI) Configuration.....                    | 1915 |
| 39.5.4         | Combined Serial Interface (CSI) Configuration.....                    | 1920 |
| 39.5.5         | Module baud rate and clock delay generation.....                      | 1922 |
| 39.5.6         | Transfer formats.....                                                 | 1925 |
| 39.5.7         | Continuous Serial Communications Clock.....                           | 1935 |
| 39.5.8         | Slave Mode Operation Constraints.....                                 | 1937 |
| 39.5.9         | Timed Serial Bus (TSB).....                                           | 1937 |
| 39.5.10        | Parity Generation and Check.....                                      | 1941 |
| 39.5.11        | Interrupts/DMA requests.....                                          | 1942 |
| 39.5.12        | Power saving features.....                                            | 1945 |
| 39.6           | Initialization/application information.....                           | 1946 |
| 39.6.1         | How to manage queues.....                                             | 1946 |
| 39.6.2         | Switching Master and Slave mode.....                                  | 1947 |
| 39.6.3         | Initializing Module in Master/Slave Modes.....                        | 1947 |
| 39.6.4         | Baud rate settings.....                                               | 1947 |
| 39.6.5         | Delay settings.....                                                   | 1948 |
| 39.6.6         | Calculation of FIFO pointer addresses.....                            | 1949 |

## Chapter 40

### Fast Ethernet Controller (FEC)

|        |                                    |      |
|--------|------------------------------------|------|
| 40.1   | Chip-specific FEC information..... | 1953 |
| 40.1.1 | FEC interface selection.....       | 1953 |
| 40.1.2 | FEC clocking.....                  | 1953 |

| Section number | Title                                                         | Page |
|----------------|---------------------------------------------------------------|------|
| 40.2           | Introduction.....                                             | 1953 |
| 40.2.1         | Block diagram.....                                            | 1953 |
| 40.2.2         | Features.....                                                 | 1955 |
| 40.2.3         | Clocking requirements.....                                    | 1956 |
| 40.3           | Modes of operation.....                                       | 1956 |
| 40.3.1         | Full and half duplex operation.....                           | 1957 |
| 40.3.2         | Interface options.....                                        | 1957 |
| 40.3.3         | Address recognition options.....                              | 1959 |
| 40.3.4         | Internal loopback.....                                        | 1959 |
| 40.4           | External signal description.....                              | 1959 |
| 40.5           | Memory map and register definition.....                       | 1960 |
| 40.5.1         | Interrupt Event Register (FEC_EIR).....                       | 1965 |
| 40.5.2         | Interrupt Mask Register (FEC_EIMR).....                       | 1967 |
| 40.5.3         | Receive Descriptor Active Register (FEC_RDAR).....            | 1969 |
| 40.5.4         | Transmit Descriptor Active Register (FEC_TDAR).....           | 1970 |
| 40.5.5         | Ethernet Control Register (FEC_ECR).....                      | 1971 |
| 40.5.6         | MII Management Frame Register (FEC_MMFR).....                 | 1972 |
| 40.5.7         | MII Speed Control Register (FEC_MSCR).....                    | 1973 |
| 40.5.8         | MIB Control Register (FEC_MIBC).....                          | 1974 |
| 40.5.9         | Receive Control Register (FEC_RCR).....                       | 1976 |
| 40.5.10        | Transmit Control Register (FEC_TCR).....                      | 1978 |
| 40.5.11        | Physical Address Low Register (FEC_PALR).....                 | 1979 |
| 40.5.12        | Physical Address High Register and Type Field (FEC_PAUR)..... | 1980 |
| 40.5.13        | Opcode/Pause Duration (FEC_OPD).....                          | 1980 |
| 40.5.14        | Descriptor Individual Upper Address Register (FEC_IAUR).....  | 1981 |
| 40.5.15        | Descriptor Individual Lower Address Register (FEC_IALR).....  | 1981 |
| 40.5.16        | Descriptor Group Upper Address Register (FEC_GAUR).....       | 1982 |
| 40.5.17        | Descriptor Group Lower Address Register (FEC_GALR).....       | 1982 |
| 40.5.18        | Transmit FIFO Watermark (FEC_TFWR).....                       | 1983 |



| Section number | Title                                                                    | Page |
|----------------|--------------------------------------------------------------------------|------|
| 40.5.19        | FIFO Receive Bound Register (FEC_FRBR).....                              | 1984 |
| 40.5.20        | FIFO Receive Start Register (FEC_FRSR).....                              | 1984 |
| 40.5.21        | Receive Descriptor Ring Start Register (FEC_ERDSR).....                  | 1985 |
| 40.5.22        | Transmit Buffer Descriptor Ring Start Register (FEC_ETDSR).....          | 1986 |
| 40.5.23        | Receive Buffer Size Register (FEC_EMRR).....                             | 1986 |
| 40.5.24        | Count of frames not counted correctly (FEC_RMON_T_DROP).....             | 1987 |
| 40.5.25        | RMON Tx packet count (FEC_RMON_T_PACKETS).....                           | 1988 |
| 40.5.26        | RMON Tx broadcast packets (FEC_RMON_T_BC_PKT).....                       | 1988 |
| 40.5.27        | RMON Tx multicast packets (FEC_RMON_T_MC_PKT).....                       | 1989 |
| 40.5.28        | RMON Tx packets with CRC/align error (FEC_RMON_T_CRC_ALIGN).....         | 1989 |
| 40.5.29        | RMON Tx packets < 64 bytes, good CRC (FEC_RMON_T_UNDERSIZE).....         | 1990 |
| 40.5.30        | RMON Tx packets > MAX_FL bytes, good CRC (FEC_RMON_T_OVERSIZE).....      | 1990 |
| 40.5.31        | RMON Tx packets < 64 bytes, bad CRC (FEC_RMON_T_FRAG).....               | 1991 |
| 40.5.32        | RMON Tx packets > MAX_FL bytes, bad CRC (FEC_RMON_T_JAB).....            | 1991 |
| 40.5.33        | RMON Tx collision count (FEC_RMON_T_COL).....                            | 1992 |
| 40.5.34        | RMON Tx 64 byte packets (FEC_RMON_T_P64).....                            | 1992 |
| 40.5.35        | RMON Tx 65 to 127 byte packets (FEC_RMON_T_P65TO127).....                | 1993 |
| 40.5.36        | RMON Tx 128 to 255 byte packets (FEC_RMON_T_P128TO255).....              | 1993 |
| 40.5.37        | RMON Tx 256 to 511 byte packets (FEC_RMON_T_P256TO511).....              | 1994 |
| 40.5.38        | RMON Tx 512 to 1023 byte packets (FEC_RMON_T_P512TO1023).....            | 1994 |
| 40.5.39        | RMON Tx 1024 to 2047 byte packets (FEC_RMON_T_P1024TO2047).....          | 1995 |
| 40.5.40        | RMON Tx packets with > 2048 bytes (FEC_RMON_T_P_GTE2048).....            | 1995 |
| 40.5.41        | RMON Tx Octets (FEC_RMON_T_OCTETS).....                                  | 1996 |
| 40.5.42        | Count of transmitted frames not counted correctly (FEC_IEEE_T_DROP)..... | 1996 |
| 40.5.43        | Frames transmitted OK (FEC_IEEE_T_FRAME_OK).....                         | 1997 |
| 40.5.44        | Frames transmitted with single collision (FEC_IEEE_T_1COL).....          | 1997 |
| 40.5.45        | Frames transmitted with multiple collisions (FEC_IEEE_T_MCOL).....       | 1998 |
| 40.5.46        | Frames transmitted after deferral delay (FEC_IEEE_T_DEF).....            | 1998 |
| 40.5.47        | Frames transmitted with late collision (FEC_IEEE_T_LCOL).....            | 1999 |

| Section number | Title                                                                        | Page |
|----------------|------------------------------------------------------------------------------|------|
| 40.5.48        | Frames transmitted with excessive collisions (FEC_IEEE_T_EXCOL).....         | 1999 |
| 40.5.49        | Frames transmitted with Tx FIFO underrun (FEC_IEEE_T_MACERR).....            | 2000 |
| 40.5.50        | Frames transmitted with carrier sense error (FEC_IEEE_T_CSERR).....          | 2000 |
| 40.5.51        | Frames transmitted with SQE error (FEC_IEEE_T_SQE).....                      | 2001 |
| 40.5.52        | Flow control pause frames transmitted (FEC_IEEE_T_FDXFC).....                | 2001 |
| 40.5.53        | Octet count for frames transmitted without error (FEC_IEEE_T_OCTETS_OK)..... | 2002 |
| 40.5.54        | Count of received frames not counted correctly (FEC_RMON_R_DROP).....        | 2002 |
| 40.5.55        | RMON Rx packet count (FEC_RMON_R_PACKETS).....                               | 2003 |
| 40.5.56        | RMON Rx broadcast packets (FEC_RMON_R_BC_PKT).....                           | 2003 |
| 40.5.57        | RMON Rx multicast packets (FEC_RMON_R_MC_PKT).....                           | 2004 |
| 40.5.58        | RMON Rx packets with CRC/Align error (FEC_RMON_R_CRC_ALIGN).....             | 2004 |
| 40.5.59        | RMON Rx packets < 64 bytes, good CRC (FEC_RMON_R_UNDERSIZE).....             | 2005 |
| 40.5.60        | RMON Rx packets > MAX_FL bytes, good CRC (FEC_RMON_R_OVERSIZE).....          | 2005 |
| 40.5.61        | RMON Rx packets < 64 bytes, bad CRC (FEC_RMON_R_FRAG).....                   | 2006 |
| 40.5.62        | RMON Rx packets > MAX_FL bytes, bad CRC (FEC_RMON_R_JAB).....                | 2006 |
| 40.5.63        | Reserved (FEC_RMON_R_RESVD_0).....                                           | 2007 |
| 40.5.64        | RMON Rx 64 byte packets (FEC_RMON_R_P64).....                                | 2007 |
| 40.5.65        | RMON Rx 65 to 127 byte packets (FEC_RMON_R_P65TO127).....                    | 2008 |
| 40.5.66        | RMON Rx 128 to 255 byte packets (FEC_RMON_R_P128TO255).....                  | 2008 |
| 40.5.67        | RMON Rx 256 to 511 byte packets (FEC_RMON_R_P256TO511).....                  | 2009 |
| 40.5.68        | RMON Rx 512 to 1023 byte packets (FEC_RMON_R_P512TO1023).....                | 2009 |
| 40.5.69        | RMON Rx 1024 to 2047 byte packets (FEC_RMON_R_P1024TO2047).....              | 2010 |
| 40.5.70        | RMON Rx packets with > 2048 bytes (FEC_RMON_R_P_GTE2048).....                | 2010 |
| 40.5.71        | RMON Rx octets (FEC_RMON_R_OCTETS).....                                      | 2011 |
| 40.5.72        | Count of received frames not counted correctly (FEC_IEEE_R_DROP).....        | 2011 |
| 40.5.73        | Frames received OK (FEC_IEEE_R_FRAME_OK).....                                | 2012 |
| 40.5.74        | Frames received with CRC error (FEC_IEEE_R_CRC).....                         | 2012 |
| 40.5.75        | Frames received with alignment error (FEC_IEEE_R_ALIGN).....                 | 2013 |
| 40.5.76        | Receive FIFO overflow count (FEC_IEEE_R_MACERR).....                         | 2013 |

| Section number | Title                                                                     | Page |
|----------------|---------------------------------------------------------------------------|------|
| 40.5.77        | Flow control pause frames received (FEC_IEEE_R_FDXFC).....                | 2014 |
| 40.5.78        | Octet count for frames received without error (FEC_IEEE_R_OCTETS_OK)..... | 2014 |
| 40.5.79        | Using the RMON and IEEE registers.....                                    | 2014 |
| 40.6           | Functional description.....                                               | 2015 |
| 40.6.1         | MII data frame.....                                                       | 2015 |
| 40.6.2         | MII management frame structure.....                                       | 2016 |
| 40.6.3         | Buffer descriptors.....                                                   | 2017 |
| 40.6.4         | Initialization sequence.....                                              | 2023 |
| 40.6.5         | User initialization (Prior to Setting ECR[ETHER_EN]).....                 | 2023 |
| 40.6.6         | Microcontroller initialization.....                                       | 2024 |
| 40.6.7         | User initialization (after setting ECR[ETHER_EN]).....                    | 2025 |
| 40.6.8         | Network interface options.....                                            | 2025 |
| 40.6.9         | FEC frame transmission.....                                               | 2026 |
| 40.6.10        | FEC frame reception.....                                                  | 2028 |
| 40.6.11        | Ethernet address recognition.....                                         | 2029 |
| 40.6.12        | Hash algorithm.....                                                       | 2032 |
| 40.6.13        | Full duplex flow control.....                                             | 2035 |
| 40.6.14        | Inter-Packet Gap (IPG) time.....                                          | 2036 |
| 40.6.15        | Collision managing.....                                                   | 2036 |
| 40.6.16        | MII internal and external loopback.....                                   | 2036 |
| 40.6.17        | RMII loopback.....                                                        | 2037 |
| 40.6.18        | RMII echo.....                                                            | 2037 |
| 40.6.19        | Ethernet error-managing procedure.....                                    | 2037 |

## Chapter 41 SENT Receiver (SRX)

|        |                                    |      |
|--------|------------------------------------|------|
| 41.1   | Chip-specific SRX information..... | 2041 |
| 41.2   | Introduction.....                  | 2041 |
| 41.2.1 | Features.....                      | 2042 |

| Section number | Title                                                                                     | Page |
|----------------|-------------------------------------------------------------------------------------------|------|
| 41.2.2         | Modes of operation.....                                                                   | 2043 |
| 41.2.3         | Block diagram.....                                                                        | 2045 |
| 41.2.4         | Design overview.....                                                                      | 2047 |
| 41.3           | External signal description.....                                                          | 2048 |
| 41.3.1         | SENT_RX [(CH-1):0].....                                                                   | 2048 |
| 41.4           | Memory map and register definition.....                                                   | 2049 |
| 41.4.1         | Global Control Register (SRX_GBL_CTRL).....                                               | 2053 |
| 41.4.2         | Channel Enable Register (SRX_CHNL_EN).....                                                | 2055 |
| 41.4.3         | Global Status Register (SRX_GBL_STATUS).....                                              | 2056 |
| 41.4.4         | Fast Message Ready Status Register (SRX_FMSG_RDY).....                                    | 2057 |
| 41.4.5         | Slow Serial Message Ready Status Register (SRX_SMSG_RDY).....                             | 2058 |
| 41.4.6         | Data Control Register 1 (SRX_DATA_CTRL1).....                                             | 2059 |
| 41.4.7         | Fast Message DMA Control Register (SRX_FDMA_CTRL).....                                    | 2061 |
| 41.4.8         | Slow Serial Message DMA Control Register (SRX_SDMA_CTRL).....                             | 2061 |
| 41.4.9         | Fast Message Ready Interrupt Control Register (SRX_FRDY_IE).....                          | 2062 |
| 41.4.10        | Slow Serial Message Ready Interrupt Enable Register (SRX_SRDY_IE).....                    | 2063 |
| 41.4.11        | DMA Fast Message Data Read Register (SRX_DMA_FMSG_DATA).....                              | 2063 |
| 41.4.12        | DMA Fast Message CRC Read Register (SRX_DMA_FMSG_CRC).....                                | 2064 |
| 41.4.13        | DMA Fast Message Time Stamp Read Register (SRX_DMA_FMSG_TS).....                          | 2065 |
| 41.4.14        | DMA Slow Serial Message Bit3 Read Register (SRX_DMA_SMSG_BIT3).....                       | 2065 |
| 41.4.15        | DMA Slow Serial Message Bit2 Read Register (SRX_DMA_SMSG_BIT2).....                       | 2067 |
| 41.4.16        | DMA Slow Serial Message Time Stamp Read Register (SRX_DMA_SMSG_TS).....                   | 2067 |
| 41.4.17        | Channel 'n' Clock Control Register (n = 0 to (CH-1)) (SRX_CHn_CLK_CTRL).....              | 2068 |
| 41.4.18        | Channel 'n' Status Register (n=0 to (CH-1)) (SRX_CHn_STATUS).....                         | 2070 |
| 41.4.19        | Channel 'n' Configuration Register (n=0 to (CH-1)) (SRX_CHn_CONFIG).....                  | 2073 |
| 41.4.20        | Channel 'n' Fast Message Data Read Register (n=0 to (CH-1)) (SRX_CHn_FMSG_DATA).....      | 2076 |
| 41.4.21        | Channel 'n' Fast Message CRC Read Register (n=0 to (CH-1)) (SRX_CHn_FMSG_CRC).....        | 2077 |
| 41.4.22        | Channel 'n' Fast Message Time Stamp Read Register (n=0 to (CH-1)) (SRX_CHn_FMSG_TS).....  | 2077 |
| 41.4.23        | Channel 'n' Serial Message Read Register (Bit 3) (n=0 to (CH-1)) (SRX_CHn_SMSG_BIT3)..... | 2078 |

| Section number | Title                                                                                      | Page |
|----------------|--------------------------------------------------------------------------------------------|------|
| 41.4.24        | Channel 'n' Serial Message Read Register (Bit 2)(n=0 to (CH-1)) (SRX_CHn_SMSG_BIT2).....   | 2079 |
| 41.4.25        | Channel 'n' Serial Message Time Stamp Read Register (n=0 to (CH-1)) (SRX_CHn_SMSG_TS)..... | 2080 |
| 41.5           | Functional description.....                                                                | 2080 |
| 41.5.1         | Initialization sequence.....                                                               | 2080 |
| 41.5.2         | DMA read logic.....                                                                        | 2081 |
| 41.5.3         | Message reading via interrupts.....                                                        | 2089 |
| 41.5.4         | Overflow behavior.....                                                                     | 2091 |
| 41.5.5         | Adjustment for variation in sensor (Tx) clock.....                                         | 2092 |
| 41.5.6         | Input programmable filter.....                                                             | 2093 |
| 41.5.7         | Receiver diagnostics.....                                                                  | 2093 |
| 41.5.8         | Time stamp logic.....                                                                      | 2100 |
| 41.5.9         | Bus Idle Diagnostic.....                                                                   | 2101 |
| 41.6           | Clocks and resets.....                                                                     | 2101 |
| 41.6.1         | Clocking strategy.....                                                                     | 2101 |
| 41.6.2         | System bus clock requirements.....                                                         | 2102 |
| 41.6.3         | High frequency receiver clock (protocol clock) requirements.....                           | 2103 |

## Chapter 42

### Enhanced Serial Communication Interface (eSCI)

|        |                                                     |      |
|--------|-----------------------------------------------------|------|
| 42.1   | Chip-specific eSCI information.....                 | 2105 |
| 42.1.1 | eSCI clocking.....                                  | 2105 |
| 42.1.2 | eSCI registers: core access in supervisor mode..... | 2105 |
| 42.2   | Introduction.....                                   | 2106 |
| 42.2.1 | Bibliography.....                                   | 2106 |
| 42.2.2 | Acronyms and Abbreviations.....                     | 2106 |
| 42.2.3 | Glossary.....                                       | 2106 |
| 42.2.4 | Overview.....                                       | 2107 |
| 42.2.5 | Features.....                                       | 2108 |

| Section number | Title                                                  | Page |
|----------------|--------------------------------------------------------|------|
| 42.2.6         | Modes of Operation.....                                | 2110 |
| 42.3           | External Signal Description.....                       | 2111 |
| 42.3.1         | Detailed Signal Descriptions.....                      | 2111 |
| 42.4           | Memory Map and Register Definition.....                | 2111 |
| 42.4.1         | Baud Rate Register (eSCI_BRR).....                     | 2112 |
| 42.4.2         | Control Register 1 (eSCI_CR1).....                     | 2113 |
| 42.4.3         | Control Register 2 (eSCI_CR2).....                     | 2115 |
| 42.4.4         | SCI Data Register (eSCI_SDR).....                      | 2117 |
| 42.4.5         | Interrupt Flag and Status Register 1 (eSCI_IFSR1)..... | 2119 |
| 42.4.6         | Interrupt Flag and Status Register 2 (eSCI_IFSR2)..... | 2121 |
| 42.4.7         | LIN Control Register 1 (eSCI_LCR1).....                | 2122 |
| 42.4.8         | LIN Control Register 2 (eSCI_LCR2).....                | 2124 |
| 42.4.9         | LIN Transmit Register (eSCI_LTR).....                  | 2124 |
| 42.4.10        | LIN Receive Register (eSCI_LRR).....                   | 2126 |
| 42.4.11        | LIN CRC Polynomial Register (eSCI_LPR).....            | 2127 |
| 42.4.12        | Control Register 3 (eSCI_CR3).....                     | 2127 |
| 42.5           | Functional Description.....                            | 2128 |
| 42.5.1         | Module Control.....                                    | 2128 |
| 42.5.2         | Frame Formats.....                                     | 2128 |
| 42.5.3         | Baud Rate and Clock Generation.....                    | 2132 |
| 42.5.4         | Baud Rate Tolerance.....                               | 2134 |
| 42.5.5         | SCI Mode.....                                          | 2136 |
| 42.5.6         | LIN Mode.....                                          | 2153 |
| 42.5.7         | Interrupts.....                                        | 2164 |
| 42.6           | Application Information.....                           | 2165 |
| 42.6.1         | SCI Data Frames Separated by Preamble.....             | 2165 |

## Chapter 43

### Resets

|      |               |      |
|------|---------------|------|
| 43.1 | Overview..... | 2167 |
|------|---------------|------|

| Section number | Title                                | Page |
|----------------|--------------------------------------|------|
| 43.2           | Reset sequence.....                  | 2168 |
| 43.3           | Reset pins.....                      | 2171 |
| 43.3.1         | RESET.....                           | 2171 |
| 43.3.2         | RSTOUT.....                          | 2171 |
| 43.4           | Reset sources.....                   | 2172 |
| 43.4.1         | Power-on reset.....                  | 2172 |
| 43.4.2         | External reset.....                  | 2174 |
| 43.4.3         | FCCU reset.....                      | 2174 |
| 43.4.4         | FOSU reset.....                      | 2175 |
| 43.4.5         | Dual core reset.....                 | 2175 |
| 43.4.6         | JTAG reset.....                      | 2175 |
| 43.4.7         | Software system reset.....           | 2176 |
| 43.4.8         | Software external reset.....         | 2176 |
| 43.4.9         | STCU reset.....                      | 2176 |
| 43.4.10        | Life cycle manipulation reset.....   | 2177 |
| 43.5           | Reset configuration.....             | 2177 |
| 43.5.1         | Weak pull up/down configuration..... | 2177 |
| 43.5.2         | Boot configuration.....              | 2178 |
| 43.6           | Boot sequence.....                   | 2178 |

## Chapter 44 System Boot

|        |                           |      |
|--------|---------------------------|------|
| 44.1   | Overview.....             | 2181 |
| 44.2   | Features.....             | 2181 |
| 44.3   | Modes of operation.....   | 2182 |
| 44.3.1 | Normal boot.....          | 2182 |
| 44.3.2 | Parallel secure boot..... | 2182 |
| 44.3.3 | Debug mode.....           | 2182 |
| 44.3.4 | Internal boot mode.....   | 2182 |
| 44.3.5 | Serial boot mode.....     | 2183 |

| Section number | Title                                     | Page |
|----------------|-------------------------------------------|------|
| 44.3.6         | Development bus boot mode.....            | 2183 |
| 44.4           | Memory Map.....                           | 2183 |
| 44.5           | Functional Description.....               | 2184 |
| 44.5.1         | BAM program flow chart.....               | 2184 |
| 44.5.2         | BAM program operation.....                | 2184 |
| 44.5.3         | Reset Configuration Half Word (RCHW)..... | 2186 |
| 44.5.4         | Internal boot mode.....                   | 2188 |
| 44.5.5         | Serial boot mode.....                     | 2189 |
| 44.5.6         | Booting from the development bus.....     | 2198 |

## Chapter 45

### System Status and Configuration Module (SSCM)

|        |                                                              |      |
|--------|--------------------------------------------------------------|------|
| 45.1   | Chip-specific SSCM information.....                          | 2201 |
| 45.1.1 | SSCM and UTEST flash memory.....                             | 2201 |
| 45.1.2 | SSCM security reset and LC state.....                        | 2202 |
| 45.1.3 | SSCM_ERROR[PAE] and abort responses to illegal accesses..... | 2203 |
| 45.2   | Introduction.....                                            | 2203 |
| 45.2.1 | Overview.....                                                | 2203 |
| 45.2.2 | Glossary.....                                                | 2204 |
| 45.2.3 | Features.....                                                | 2204 |
| 45.2.4 | Modes of operation.....                                      | 2205 |
| 45.3   | External signal description.....                             | 2205 |
| 45.4   | Memory map and register definition.....                      | 2205 |
| 45.4.1 | SSCM System Status (SSCM_STATUS).....                        | 2206 |
| 45.4.2 | SSCM Error Configuration Register (SSCM_ERROR).....          | 2208 |
| 45.4.3 | Life Cycle Status Register (SSCM_LCSTAT).....                | 2209 |
| 45.5   | Functional description.....                                  | 2209 |
| 45.5.1 | DCF mechanism.....                                           | 2210 |
| 45.5.2 | Life Cycle.....                                              | 2212 |



| Section number | Title                                           | Page |
|----------------|-------------------------------------------------|------|
| 45.6           | Initialization and application information..... | 2214 |
| 45.6.1         | Reset.....                                      | 2214 |
| 45.7           | Additional safety measures.....                 | 2214 |
| 45.7.1         | Spurious reset protection.....                  | 2214 |

## Chapter 46 Power Management

|      |                                          |      |
|------|------------------------------------------|------|
| 46.1 | Overview.....                            | 2215 |
| 46.2 | Functional description.....              | 2216 |
| 46.3 | Power modes.....                         | 2216 |
| 46.4 | External supply.....                     | 2216 |
| 46.5 | Power management supply description..... | 2216 |
| 46.6 | Voltage monitoring.....                  | 2217 |

## Chapter 47 Power Management Controller (PMC)

|         |                                                                       |      |
|---------|-----------------------------------------------------------------------|------|
| 47.1    | Introduction.....                                                     | 2219 |
| 47.1.1  | Features.....                                                         | 2219 |
| 47.2    | Memory Map and Registers.....                                         | 2221 |
| 47.2.1  | Temperature Sensor Interrupt Enable Register (PMC_TS_IER).....        | 2222 |
| 47.2.2  | Temperature Event Status Register (PMC_ESR_TD).....                   | 2224 |
| 47.2.3  | Temperature Reset Event Enable register (PMC_REE_TD).....             | 2226 |
| 47.2.4  | Temperature Reset Event Selection Register (PMC_RES_TD).....          | 2228 |
| 47.2.5  | Temperature Detector Configuration Register (PMC_CTL_TD).....         | 2229 |
| 47.2.6  | Temperature Sensor Fault Injection Register (PMC_TS_FIR).....         | 2230 |
| 47.2.7  | Voltage Detect User Mode Test Register (PMC_SELF_TEST_UM_VD_REG)..... | 2231 |
| 47.2.8  | FCCU Fault Injection Register (PMC_FIR).....                          | 2234 |
| 47.2.9  | LVD/HVD Interrupt Enable Control Register (PMC_LVD_HVD_INTR_ENB)..... | 2235 |
| 47.2.10 | LVD/HVD Event Status Register (PMC_LVD_HVD_EVENT_STATUS).....         | 2238 |
| 47.2.11 | ADC Channel Select Register (PMC_ADC_CS).....                         | 2240 |
| 47.2.12 | Reset Event Enable Register (PMC_REE).....                            | 2242 |

| Section number | Title                                   | Page |
|----------------|-----------------------------------------|------|
| 47.3           | Functional description.....             | 2244 |
| 47.3.1         | Analog PMC interface.....               | 2244 |
| 47.3.2         | Temperature Sensor interface logic..... | 2244 |
| 47.3.3         | POR phase gates.....                    | 2244 |
| 47.3.4         | Flash memory interface to DCF.....      | 2244 |
| 47.3.5         | FCCU interface.....                     | 2245 |
| 47.3.6         | LVD/HVD self test mechanism.....        | 2245 |

## Chapter 48 JTAG Controller (JTAGC)

|        |                                      |      |
|--------|--------------------------------------|------|
| 48.1   | Chip-specific JTAGC information..... | 2249 |
| 48.1.1 | JTAG ID.....                         | 2249 |
| 48.2   | Introduction.....                    | 2249 |
| 48.2.1 | Block diagram.....                   | 2249 |
| 48.2.2 | Features.....                        | 2250 |
| 48.2.3 | Modes of operation.....              | 2250 |
| 48.3   | External signal description.....     | 2252 |
| 48.3.1 | TCK—Test clock input.....            | 2252 |
| 48.3.2 | TDI—Test data input.....             | 2252 |
| 48.3.3 | TDO—Test data output.....            | 2252 |
| 48.3.4 | TMS—Test mode select.....            | 2253 |
| 48.3.5 | JCOMP—JTAG compliancy.....           | 2253 |
| 48.4   | Register description.....            | 2253 |
| 48.4.1 | Instruction register.....            | 2253 |
| 48.4.2 | Bypass register.....                 | 2254 |
| 48.4.3 | Device identification register.....  | 2254 |
| 48.4.4 | JTAG_PASSWORD register.....          | 2255 |
| 48.4.5 | Boundary scan register.....          | 2255 |
| 48.5   | Functional description.....          | 2256 |
| 48.5.1 | JTAGC reset configuration.....       | 2256 |

| Section number | Title                                         | Page |
|----------------|-----------------------------------------------|------|
| 48.5.2         | IEEE 1149.1-2001 (JTAG) Test Access Port..... | 2256 |
| 48.5.3         | TAP controller state machine.....             | 2256 |
| 48.5.4         | JTAGC block instructions.....                 | 2258 |
| 48.5.5         | Boundary scan.....                            | 2261 |
| 48.6           | Initialization/Application information.....   | 2262 |

## Chapter 49

### IEEE 1149.7 Compact JTAG Test Access Port Controller (CJTAG)

|        |                                   |      |
|--------|-----------------------------------|------|
| 49.1   | References.....                   | 2263 |
| 49.2   | Abbreviations.....                | 2263 |
| 49.3   | Introduction.....                 | 2264 |
| 49.3.1 | Types of operation.....           | 2265 |
| 49.3.2 | Deployment by class.....          | 2265 |
| 49.3.3 | 1149.7 TAP signals.....           | 2266 |
| 49.3.4 | TAP.7 architecture.....           | 2267 |
| 49.3.5 | Protocols.....                    | 2268 |
| 49.4   | Operating models.....             | 2270 |
| 49.5   | CJTAG implementation summary..... | 2270 |
| 49.5.1 | T0 functions.....                 | 2270 |
| 49.5.2 | T1 functions.....                 | 2271 |
| 49.5.3 | T2 functions.....                 | 2271 |
| 49.5.4 | T3 functions.....                 | 2271 |
| 49.5.5 | T4 functions.....                 | 2271 |
| 49.6   | Ancillary services.....           | 2271 |
| 49.6.1 | Overview.....                     | 2271 |
| 49.6.2 | Resets.....                       | 2272 |
| 49.6.3 | Start-up Options.....             | 2275 |
| 49.6.4 | RSU operation.....                | 2276 |
| 49.6.5 | TAPC State Machine.....           | 2278 |

| Section number | Title                                                      | Page |
|----------------|------------------------------------------------------------|------|
| 49.7           | EPU (Extended Protocol Unit) Operation.....                | 2279 |
| 49.7.1         | EPU Operation.....                                         | 2279 |
| 49.7.2         | EPU Registers.....                                         | 2282 |
| 49.7.3         | EPU Commands.....                                          | 2293 |
| 49.7.4         | EPU operating states.....                                  | 2301 |
| 49.7.5         | System and EPU Paths.....                                  | 2302 |
| 49.8           | APU (Advanced Protocol Unit) Operation.....                | 2302 |
| 49.8.1         | Overview.....                                              | 2302 |
| 49.8.2         | Operation.....                                             | 2305 |
| 49.8.3         | Escape sequences.....                                      | 2307 |
| 49.8.4         | Signal behaviors.....                                      | 2307 |
| 49.8.5         | APU Functions.....                                         | 2308 |
| 49.8.6         | Configuration Change Packets (CP).....                     | 2314 |
| 49.8.7         | Scan Packet.....                                           | 2315 |
| 49.8.8         | SP Format.....                                             | 2315 |
| 49.9           | Functional Description.....                                | 2322 |
| 49.9.1         | Switching from Standard Protocol to Advanced Protocol..... | 2322 |

## Chapter 50 JTAG Data Communication (JDC)

|        |                                               |      |
|--------|-----------------------------------------------|------|
| 50.1   | Introduction.....                             | 2323 |
| 50.2   | Overview.....                                 | 2323 |
| 50.3   | Memory mapped registers.....                  | 2324 |
| 50.3.1 | Module Configuration Register (JDC_MCR).....  | 2325 |
| 50.3.2 | Module Status Register (JDC_MSR).....         | 2326 |
| 50.3.3 | JTAG Output Data Register (JDC_JOUT_IPS)..... | 2327 |
| 50.3.4 | JTAG Input Data Register (JDC_JIN_IPS).....   | 2328 |
| 50.4   | Non-Memory mapped register definition.....    | 2328 |
| 50.4.1 | JTAG output data register (JOUT).....         | 2328 |
| 50.4.2 | JTAG input data register (JIN).....           | 2329 |

| Section number | Title                       | Page |
|----------------|-----------------------------|------|
| 50.5           | Functional description..... | 2329 |
| 50.5.1         | JTAG register access.....   | 2330 |

## Chapter 51 Nexus Development Interface (NDI)

|        |                                                                  |      |
|--------|------------------------------------------------------------------|------|
| 51.1   | Overview.....                                                    | 2331 |
| 51.2   | Nexus clients for TAP access.....                                | 2332 |
| 51.3   | Debug break and cross-triggering.....                            | 2332 |
| 51.3.1 | Break control.....                                               | 2333 |
| 51.3.2 | Cross-triggering control.....                                    | 2333 |
| 51.3.3 | Synchronous restart control.....                                 | 2333 |
| 51.4   | Nexus eDMA Interface (NXDM) and Nexus SIPI Interface (NXSI)..... | 2334 |
| 51.4.1 | Introduction.....                                                | 2334 |
| 51.4.2 | External Signal Description.....                                 | 2336 |
| 51.4.3 | NXDM and NXSI Programmers Model.....                             | 2336 |
| 51.4.4 | Functional Description.....                                      | 2347 |

## Chapter 52 Nexus Port Controller (NPC)

|        |                                    |      |
|--------|------------------------------------|------|
| 52.1   | Chip-specific NPC information..... | 2357 |
| 52.1.1 | DID register reset value.....      | 2357 |
| 52.1.2 | NPC MDO pins.....                  | 2357 |
| 52.2   | Introduction.....                  | 2358 |
| 52.2.1 | Overview.....                      | 2358 |
| 52.2.2 | Features.....                      | 2358 |
| 52.2.3 | Modes of operation.....            | 2359 |
| 52.3   | External signal description.....   | 2361 |
| 52.3.1 | Overview.....                      | 2361 |

| Section number | Title                                       | Page |
|----------------|---------------------------------------------|------|
| 52.3.2         | Detailed signal descriptions.....           | 2361 |
| 52.4           | Register definition.....                    | 2363 |
| 52.4.1         | Register descriptions.....                  | 2363 |
| 52.5           | Functional Description.....                 | 2368 |
| 52.5.1         | NPC reset configuration.....                | 2368 |
| 52.5.2         | Auxiliary Output Port.....                  | 2368 |
| 52.5.3         | IEEE 1149.1-2001 (JTAG) TAP.....            | 2371 |
| 52.5.4         | Nexus JTAG Port Sharing.....                | 2376 |
| 52.5.5         | MCKO Control .....                          | 2376 |
| 52.5.6         | EVTO Sharing.....                           | 2377 |
| 52.5.7         | Nexus Reset Control.....                    | 2377 |
| 52.5.8         | System Clock Locked Indication.....         | 2377 |
| 52.6           | Initialization/Application Information..... | 2378 |
| 52.6.1         | Accessing NPC tool-mapped registers.....    | 2378 |

## Chapter 53

### Nexus eTPU Development Interface (NDEDI)

|        |                                             |      |
|--------|---------------------------------------------|------|
| 53.1   | Chip-specific NDEDI information.....        | 2379 |
| 53.1.1 | NDEDI/NSEDI device-specific parameters..... | 2379 |
| 53.2   | Introduction.....                           | 2380 |
| 53.3   | Features.....                               | 2381 |
| 53.4   | Modes of operation.....                     | 2383 |
| 53.4.1 | Reset.....                                  | 2383 |
| 53.4.2 | Disabled-port mode.....                     | 2383 |
| 53.4.3 | Full-port mode.....                         | 2383 |
| 53.4.4 | Reduced-port mode.....                      | 2384 |
| 53.4.5 | Stop mode.....                              | 2384 |
| 53.5   | External signal description.....            | 2384 |
| 53.5.1 | Overview.....                               | 2384 |

| Section number | Title                                       | Page |
|----------------|---------------------------------------------|------|
| 53.5.2         | Detailed signal descriptions.....           | 2385 |
| 53.6           | Register definition.....                    | 2387 |
| 53.6.1         | Register descriptions.....                  | 2389 |
| 53.6.2         | Unimplemented Registers.....                | 2421 |
| 53.7           | Functional description.....                 | 2421 |
| 53.7.1         | NDEDI reset configuration.....              | 2421 |
| 53.7.2         | Auxiliary output port.....                  | 2422 |
| 53.7.3         | IEEE 1149.1 (JTAG) Input Port.....          | 2432 |
| 53.7.4         | Microcode Development Support.....          | 2437 |
| 53.7.5         | Debug Status.....                           | 2438 |
| 53.7.6         | Ownership Trace.....                        | 2440 |
| 53.7.7         | Program Trace.....                          | 2441 |
| 53.7.8         | Data Trace.....                             | 2456 |
| 53.7.9         | Watchpoint Trace.....                       | 2463 |
| 53.7.10        | ETPU Message Queue.....                     | 2465 |
| 53.8           | Initialization/Application Information..... | 2469 |
| 53.8.1         | Accessing NDEDI tool-mapped registers.....  | 2469 |
| 53.8.2         | Program Trace Reconstruction.....           | 2470 |
| 53.8.3         | Microcode Development Support.....          | 2474 |

## Chapter 54 Development Trigger Semaphore (DTS)

|        |                                          |      |
|--------|------------------------------------------|------|
| 54.1   | Introduction.....                        | 2479 |
| 54.2   | Overview.....                            | 2479 |
| 54.3   | DTS device connections.....              | 2481 |
| 54.3.1 | DTS register access.....                 | 2482 |
| 54.4   | Memory mapped registers.....             | 2483 |
| 54.4.1 | Output Enable Register (DTS_ENABLE)..... | 2483 |
| 54.4.2 | Startup Register (DTS_STARTUP).....      | 2484 |
| 54.4.3 | Semaphore Register (DTS_SEMAPHORE).....  | 2485 |

| Section number | Title                                      | Page |
|----------------|--------------------------------------------|------|
| 54.4.4         | Semaphore Extension (DTS_SEMAPHORE_B)..... | 2486 |
| 54.5           | Example application.....                   | 2486 |

## Chapter 55

### Fault Collection and Control Unit (FCCU)

|        |                                                                              |      |
|--------|------------------------------------------------------------------------------|------|
| 55.1   | Chip-specific FCCU information.....                                          | 2489 |
| 55.1.1 | FCCU register default values.....                                            | 2489 |
| 55.1.2 | FOSU_COUNT value.....                                                        | 2489 |
| 55.1.3 | FCCU fault inputs.....                                                       | 2490 |
| 55.1.4 | Error signal flow diagram.....                                               | 2494 |
| 55.1.5 | Restriction on configuring RESET and NMI reactions on same FCCU channel..... | 2495 |
| 55.1.6 | FCCU_CFG register event bit values by source (N and C).....                  | 2496 |
| 55.2   | Introduction.....                                                            | 2497 |
| 55.2.1 | Acronyms and abbreviations.....                                              | 2498 |
| 55.3   | Main features.....                                                           | 2499 |
| 55.4   | Block diagram.....                                                           | 2499 |
| 55.5   | Signal description.....                                                      | 2501 |
| 55.5.1 | Signals.....                                                                 | 2501 |
| 55.6   | Put FCCU in Configuration or Normal state.....                               | 2502 |
| 55.6.1 | Introduction.....                                                            | 2502 |
| 55.6.2 | About changing states.....                                                   | 2503 |
| 55.6.3 | About locking the configuration.....                                         | 2503 |
| 55.6.4 | Put FCCU in Configuration state.....                                         | 2503 |
| 55.6.5 | Put FCCU in Normal state.....                                                | 2504 |
| 55.7   | Run operations.....                                                          | 2505 |
| 55.7.1 | Introduction.....                                                            | 2505 |
| 55.7.2 | About running operations.....                                                | 2505 |
| 55.7.3 | Run an operation.....                                                        | 2505 |
| 55.8   | Functional description.....                                                  | 2506 |
| 55.8.1 | Definitions.....                                                             | 2506 |



| Section number | Title                                                       | Page |
|----------------|-------------------------------------------------------------|------|
| 55.8.2         | FSM description.....                                        | 2506 |
| 55.8.3         | Fault priority scheme and nesting.....                      | 2509 |
| 55.8.4         | STCU interface.....                                         | 2510 |
| 55.8.5         | Nonvolatile memory interface.....                           | 2511 |
| 55.8.6         | EOUT interface.....                                         | 2512 |
| 55.8.7         | Error signal flow.....                                      | 2519 |
| 55.9           | Register descriptions.....                                  | 2519 |
| 55.9.1         | Control (FCCU_CTRL).....                                    | 2522 |
| 55.9.2         | Control Key (FCCU_CTRLK).....                               | 2524 |
| 55.9.3         | Configuration (FCCU_CFG).....                               | 2525 |
| 55.9.4         | Noncritical Fault Configuration (FCCU_NCF_CFGn).....        | 2528 |
| 55.9.5         | Noncritical Fault State Configuration (FCCU_NCFS_CFGn)..... | 2529 |
| 55.9.6         | Noncritical Fault Status (FCCU_NCF_Sn).....                 | 2530 |
| 55.9.7         | Noncritical Fault Key (FCCU_NCFK).....                      | 2532 |
| 55.9.8         | Noncritical Fault Enable (FCCU_NCF_En).....                 | 2533 |
| 55.9.9         | Noncritical Fault Timeout Enable (FCCU_NCF_TOEn).....       | 2534 |
| 55.9.10        | Noncritical Fault Timeout (FCCU_NCF_TO).....                | 2535 |
| 55.9.11        | Configuration-State Timer Interval (FCCU_CFG_TO).....       | 2535 |
| 55.9.12        | IO Control (FCCU_EINOUT).....                               | 2536 |
| 55.9.13        | Status (FCCU_STAT).....                                     | 2538 |
| 55.9.14        | NA Freeze Status (FCCU_N2AF_STATUS).....                    | 2540 |
| 55.9.15        | AF Freeze Status (FCCU_A2FF_STATUS).....                    | 2541 |
| 55.9.16        | NF Freeze Status (FCCU_N2FF_STATUS).....                    | 2542 |
| 55.9.17        | FA Freeze Status (FCCU_F2A_STATUS).....                     | 2543 |
| 55.9.18        | Noncritical Fault Fake (FCCU_NCF_F).....                    | 2544 |
| 55.9.19        | IRQ Status (FCCU_IRQ_STAT).....                             | 2545 |
| 55.9.20        | IRQ Enable (FCCU_IRQ_EN).....                               | 2546 |
| 55.9.21        | X Timer (FCCU_XTMR).....                                    | 2547 |
| 55.9.22        | Transient Configuration Lock (FCCU_TRANS_LOCK).....         | 2548 |

| Section number | Title                                                                               | Page |
|----------------|-------------------------------------------------------------------------------------|------|
| 55.9.23        | Permanent Configuration Lock (FCCU_PERMNT_LOCK).....                                | 2549 |
| 55.9.24        | Delta T (FCCU_DELTA_T).....                                                         | 2549 |
| 55.9.25        | IRQ Alarm Enable (FCCU_IRQ_ALARM_EN <sub>n</sub> ).....                             | 2550 |
| 55.9.26        | NMI Enable (FCCU_NMI_EN <sub>n</sub> ).....                                         | 2551 |
| 55.9.27        | Noncritical Fault-State EOUT Signaling Enable (FCCU_EOUT_SIG_EN <sub>n</sub> )..... | 2552 |
| 55.9.28        | Configuration registers.....                                                        | 2553 |
| 55.9.29        | FCCU_CFG register bit value sources (N and C) by event.....                         | 2553 |
| 55.10          | FCCU Output Supervision Unit.....                                                   | 2555 |
| 55.11          | Use cases and limitations.....                                                      | 2557 |
| 55.11.1        | Configuration guidelines.....                                                       | 2557 |
| 55.11.2        | Recommendations to configure FCCU.....                                              | 2558 |

## Chapter 56

### Self-Test Control Unit (STCU2)

|         |                                                                                                 |      |
|---------|-------------------------------------------------------------------------------------------------|------|
| 56.1    | Chip-specific STCU information.....                                                             | 2559 |
| 56.1.1  | Supported BIST sequences.....                                                                   | 2559 |
| 56.1.2  | Configure or bypass STCU2 off-line self-tests.....                                              | 2560 |
| 56.1.3  | STCU interfaces to other modules.....                                                           | 2560 |
| 56.1.4  | On-line LBIST and MBIST clock configuration.....                                                | 2561 |
| 56.1.5  | STCU2_PLL_CFG configuration of PLLDIG_PLL0DV.....                                               | 2562 |
| 56.1.6  | Wait time for writing to the on-line registers.....                                             | 2562 |
| 56.1.7  | On-line MBIST.....                                                                              | 2562 |
| 56.1.8  | LBIST scenarios.....                                                                            | 2565 |
| 56.1.9  | STCU register default values.....                                                               | 2569 |
| 56.1.10 | AUTOLOCK_VALUE for register write access and DCF_COMPLETION value for DCF write completion..... | 2569 |
| 56.2    | Introduction.....                                                                               | 2570 |
| 56.3    | Main features.....                                                                              | 2570 |
| 56.4    | Block diagram.....                                                                              | 2572 |
| 56.5    | IPS bus interface.....                                                                          | 2573 |

| Section number | Title                                                 | Page |
|----------------|-------------------------------------------------------|------|
| 56.6           | BISTs and BIST partitions.....                        | 2574 |
| 56.6.1         | Definition: BIST.....                                 | 2574 |
| 56.6.2         | Definition: BIST partition.....                       | 2574 |
| 56.6.3         | Example: BIST partitions on a chip.....               | 2575 |
| 56.6.4         | Types of BIST partitions.....                         | 2575 |
| 56.7           | BIST sequences.....                                   | 2575 |
| 56.7.1         | Definition: BIST sequence.....                        | 2575 |
| 56.7.2         | STCU2 executes MBISTs before LBISTs.....              | 2575 |
| 56.7.3         | Example: Single-phase BIST sequence.....              | 2576 |
| 56.7.4         | Example: Multiphase BIST sequence.....                | 2576 |
| 56.7.5         | Types of BIST sequences.....                          | 2576 |
| 56.7.6         | Supported BIST sequences.....                         | 2577 |
| 56.7.7         | Definition: Default offline BIST sequence.....        | 2577 |
| 56.8           | Functional description.....                           | 2577 |
| 56.8.1         | FSM description.....                                  | 2577 |
| 56.8.2         | Reset management.....                                 | 2578 |
| 56.8.3         | Built-in self-test scheduling.....                    | 2578 |
| 56.8.4         | PLL interface.....                                    | 2579 |
| 56.8.5         | Interrupt interface.....                              | 2579 |
| 56.8.6         | FCCU interface.....                                   | 2580 |
| 56.8.7         | Watchdogs.....                                        | 2580 |
| 56.8.8         | SSCM interface.....                                   | 2581 |
| 56.9           | Register description.....                             | 2582 |
| 56.9.1         | STCU2 Run Register (STCU2_RUN).....                   | 2589 |
| 56.9.2         | STCU2 Run Software Register (STCU2_RUNSW).....        | 2590 |
| 56.9.3         | STCU2 SK Code Register (STCU2_SKC).....               | 2591 |
| 56.9.4         | STCU2 Configuration Register (STCU2_CFG).....         | 2593 |
| 56.9.5         | STCU2 PLL Configuration Register (STCU2_PLL_CFG)..... | 2596 |
| 56.9.6         | STCU2 Watchdog Granularity (STCU2_WDG).....           | 2597 |

| Section number | Title                                                                        | Page |
|----------------|------------------------------------------------------------------------------|------|
| 56.9.7         | STCU2 Interrupt Flag Register (STCU2_INT_FLG).....                           | 2599 |
| 56.9.8         | STCU2 Error Register (STCU2_ERR_STAT).....                                   | 2600 |
| 56.9.9         | STCU2 Error FM Register (STCU2_ERR_FM).....                                  | 2603 |
| 56.9.10        | STCU2 Off-Line LBIST Status Register (STCU2_LBS).....                        | 2604 |
| 56.9.11        | STCU2 Off-Line LBIST End Flag Register (STCU2_LBE).....                      | 2605 |
| 56.9.12        | STCU2 On-Line LBIST Status Register (STCU2_LBSSW).....                       | 2606 |
| 56.9.13        | STCU2 On-Line LBIST End Flag Register (STCU2_LBESW).....                     | 2608 |
| 56.9.14        | STCU2 On-Line LBIST Reset Management (STCU2_LBRMSW).....                     | 2609 |
| 56.9.15        | STCU2 LBIST Unrecoverable FM Register (STCU2_LBUFM).....                     | 2611 |
| 56.9.16        | STCU2 Off-Line MBIST Status Low Register (STCU2_MBSL).....                   | 2612 |
| 56.9.17        | STCU2 Off-Line MBIST Status Medium Register (STCU2_MBSM).....                | 2617 |
| 56.9.18        | STCU2 Off-Line MBIST Status High Register (STCU2_MBSH).....                  | 2622 |
| 56.9.19        | STCU2 Off-Line MBIST End Flag Low Register (STCU2_MBEL).....                 | 2623 |
| 56.9.20        | STCU2 Off-Line MBIST End Flag Medium Register (STCU2_MBEM).....              | 2627 |
| 56.9.21        | STCU2 Off-Line MBIST End Flag High Register (STCU2_MBEH).....                | 2631 |
| 56.9.22        | STCU2 On-Line MBIST Status Low Register (STCU2_MBSLSW).....                  | 2632 |
| 56.9.23        | STCU2 On-Line MBIST Status Medium Register (STCU2_MBSMSW).....               | 2635 |
| 56.9.24        | STCU2 On-Line MBIST Status High Register (STCU2_MBSHSW).....                 | 2639 |
| 56.9.25        | STCU2 On-Line MBIST End Flag Low Register (STCU2_MBELSW).....                | 2640 |
| 56.9.26        | STCU2 On-Line MBIST End Flag Medium Register (STCU2_MBEMSW).....             | 2644 |
| 56.9.27        | STCU2 On-Line MBIST End Flag High Register (STCU2_MBEHSW).....               | 2648 |
| 56.9.28        | STCU2 MBIST Unrecoverable FM Low Register (STCU2_MBUFML).....                | 2649 |
| 56.9.29        | STCU2 MBIST Unrecoverable FM Medium Register (STCU2_MBUFMM).....             | 2652 |
| 56.9.30        | STCU2 MBIST Unrecoverable FM High Register (STCU2_MBUFMH).....               | 2656 |
| 56.9.31        | STCU2 LBIST Control (STCU2_LB_CTRL $n$ ).....                                | 2657 |
| 56.9.32        | STCU2 LBIST PC Stop Register (STCU2_LB_PCS $n$ ).....                        | 2660 |
| 56.9.33        | STCU2 Off-Line LBIST MISR Expected Low Register (STCU2_LB_MISREL $n$ ).....  | 2660 |
| 56.9.34        | STCU2 Off-Line LBIST MISR Expected High Register (STCU2_LB_MISREH $n$ )..... | 2661 |
| 56.9.35        | STCU2 Off-Line LBIST MISR Read Low Register (STCU2_LB_MISRRL $n$ ).....      | 2662 |

| Section number | Title                                                                                  | Page |
|----------------|----------------------------------------------------------------------------------------|------|
| 56.9.36        | STCU2 Off-Line LBIST MISR Read High Register (STCU2_LB_MISRRH <sub>n</sub> ).....      | 2662 |
| 56.9.37        | STCU2 On-Line LBIST MISR Expected Low Register (STCU2_LB_MISRELSW <sub>n</sub> ).....  | 2663 |
| 56.9.38        | STCU2 On-Line LBIST MISR Expected High Register (STCU2_LB_MISREHSW <sub>n</sub> )..... | 2664 |
| 56.9.39        | STCU2 On-Line LBIST MISR Read Low Register (STCU2_LB_MISRRLSW <sub>n</sub> ).....      | 2664 |
| 56.9.40        | STCU2 On-Line LBIST MISR Read High Register (STCU2_LB_MISRRHSW <sub>n</sub> ).....     | 2665 |
| 56.9.41        | STCU2 MBIST Control Register (STCU2_MB_CTRL <sub>n</sub> ).....                        | 2666 |
| 56.10          | Use cases and limitations.....                                                         | 2667 |
| 56.10.1        | Offline self-test sequence.....                                                        | 2667 |
| 56.10.2        | Online self test sequence.....                                                         | 2669 |
| 56.10.3        | Bypass USER Mode.....                                                                  | 2671 |
| 56.10.4        | Design implementation information.....                                                 | 2671 |

## Chapter 57

### Cyclic Redundancy Check (CRC) Unit

|        |                                                        |      |
|--------|--------------------------------------------------------|------|
| 57.1   | Chip-specific CRC information.....                     | 2673 |
| 57.2   | Introduction.....                                      | 2673 |
| 57.3   | Main features.....                                     | 2673 |
| 57.3.1 | Standard features.....                                 | 2673 |
| 57.4   | Block diagram.....                                     | 2674 |
| 57.5   | External signal description.....                       | 2674 |
| 57.5.1 | Peripheral bus interface.....                          | 2674 |
| 57.6   | CRC memory map and registers.....                      | 2675 |
| 57.6.1 | Configuration Register (CRC_CFG <sub>n</sub> ).....    | 2676 |
| 57.6.2 | Input Register (CRC_INP <sub>n</sub> ).....            | 2677 |
| 57.6.3 | Current Status Register (CRC_CSTAT <sub>n</sub> )..... | 2678 |
| 57.6.4 | Output Register (CRC_OUTP <sub>n</sub> ).....          | 2679 |
| 57.7   | Functional description.....                            | 2679 |
| 57.8   | Use cases.....                                         | 2681 |
| 57.8.1 | Programming example.....                               | 2681 |
| 57.8.2 | Register programming.....                              | 2683 |

| Section number                             | Title                                                         | Page |
|--------------------------------------------|---------------------------------------------------------------|------|
| <b>Chapter 58</b>                          |                                                               |      |
| <b>Security Overview</b>                   |                                                               |      |
| 58.1                                       | Introduction.....                                             | 2685 |
| 58.2                                       | Security features summary.....                                | 2685 |
| 58.3                                       | Flash memory read, write, and erase protection.....           | 2686 |
| <b>Chapter 59</b>                          |                                                               |      |
| <b>Cryptographic Services Engine (CSE)</b> |                                                               |      |
| 59.1                                       | Chip-specific CSE information.....                            | 2687 |
| 59.1.1                                     | CSE configuration and SECURE_BOOT execution.....              | 2687 |
| 59.1.2                                     | CSE Clk Freq.....                                             | 2687 |
| 59.1.3                                     | CSE clock: minimum frequency.....                             | 2687 |
| 59.1.4                                     | CSE debug not supported.....                                  | 2688 |
| 59.1.5                                     | CSE command-processing completion before Stop mode entry..... | 2688 |
| 59.2                                       | Introduction.....                                             | 2688 |
| 59.2.1                                     | Overview.....                                                 | 2688 |
| 59.2.2                                     | Features.....                                                 | 2688 |
| 59.2.3                                     | Modes of operation.....                                       | 2689 |
| 59.2.4                                     | Block diagram.....                                            | 2689 |
| 59.3                                       | External signal description.....                              | 2689 |
| 59.4                                       | Memory Map and Register Definition.....                       | 2690 |
| 59.4.1                                     | CSE Control Register (CSE_CR).....                            | 2690 |
| 59.4.2                                     | CSE Status Register (CSE_SR).....                             | 2692 |
| 59.4.3                                     | CSE Interrupt Register (CSE_IR).....                          | 2693 |
| 59.4.4                                     | CSE Error Code Register (CSE_ECR).....                        | 2694 |
| 59.4.5                                     | CSE Command Register (CSE_CMD).....                           | 2695 |
| 59.4.6                                     | CSE Parameter Register (CSE_Pn).....                          | 2696 |
| 59.5                                       | CSE functional description.....                               | 2696 |
| 59.5.1                                     | Host interface.....                                           | 2696 |
| 59.5.2                                     | Command processing.....                                       | 2697 |

| Section number | Title                                   | Page |
|----------------|-----------------------------------------|------|
| 59.5.3         | Secure storage.....                     | 2698 |
| 59.5.4         | Encryption and decryption.....          | 2700 |
| 59.5.5         | Message authentication.....             | 2700 |
| 59.5.6         | Secure boot.....                        | 2701 |
| 59.5.7         | Random number generation.....           | 2702 |
| 59.5.8         | Error handling.....                     | 2702 |
| 59.6           | CSE commands.....                       | 2704 |
| 59.6.1         | Encrypt ECB.....                        | 2704 |
| 59.6.2         | Encrypt CBC.....                        | 2705 |
| 59.6.3         | Decrypt ECB.....                        | 2705 |
| 59.6.4         | Decrypt CBC.....                        | 2705 |
| 59.6.5         | Generate MAC.....                       | 2706 |
| 59.6.6         | Verify MAC.....                         | 2706 |
| 59.6.7         | Load Key.....                           | 2707 |
| 59.6.8         | Load Plain Key.....                     | 2708 |
| 59.6.9         | Export RAM Key.....                     | 2708 |
| 59.6.10        | Initialize RNG.....                     | 2708 |
| 59.6.11        | Extend PRNG Seed.....                   | 2709 |
| 59.6.12        | Generate Random Number.....             | 2709 |
| 59.6.13        | Secure Boot.....                        | 2710 |
| 59.6.14        | Boot Failure.....                       | 2710 |
| 59.6.15        | Boot OK.....                            | 2710 |
| 59.6.16        | Get ID.....                             | 2711 |
| 59.6.17        | Cancel.....                             | 2711 |
| 59.6.18        | Debug Challenge.....                    | 2711 |
| 59.6.19        | Debug Authorization.....                | 2712 |
| 59.6.20        | Generate TRNG Random Number.....        | 2712 |
| 59.6.21        | Initialize CSE.....                     | 2713 |
| 59.6.22        | Miyaguchi-Preneel (MP) Compression..... | 2713 |

## Chapter 60

### Password and Device Security Module (PASS)

|        |                                                                        |      |
|--------|------------------------------------------------------------------------|------|
| 60.1   | Chip-specific PASS information.....                                    | 2715 |
| 60.1.1 | Accesses to unimplemented PASS registers.....                          | 2715 |
| 60.1.2 | PASS_LOCKm_PGn register bit mapping.....                               | 2715 |
| 60.1.3 | PASS chip select (client select).....                                  | 2719 |
| 60.1.4 | Flash memory read protection scenarios.....                            | 2719 |
| 60.1.5 | Debug Interface Access.....                                            | 2720 |
| 60.1.6 | Duration of Debug Interface Access via JTAG password.....              | 2721 |
| 60.1.7 | Production Disable enablement.....                                     | 2721 |
| 60.1.8 | PASS_LCSTAT[LIFE] value source.....                                    | 2722 |
| 60.2   | Introduction.....                                                      | 2722 |
| 60.2.1 | Overview.....                                                          | 2722 |
| 60.2.2 | Features.....                                                          | 2722 |
| 60.2.3 | Modes of operation.....                                                | 2723 |
| 60.3   | Memory map and register definition.....                                | 2723 |
| 60.3.1 | Life Cycle Status Register (PASS_LCSTAT).....                          | 2725 |
| 60.3.2 | Challenge Selector Register (PASS_CHSEL).....                          | 2726 |
| 60.3.3 | Challenge Status Register (PASS_CSTAT).....                            | 2727 |
| 60.3.4 | Challenge Input Register (PASS_CINn).....                              | 2727 |
| 60.3.5 | Password Group <i>n</i> - Lock 0 Status Register (PASS_LOCK0_PGn)..... | 2728 |
| 60.3.6 | Password Group <i>n</i> - Lock 1 Status Register (PASS_LOCK1_PGn)..... | 2730 |
| 60.3.7 | Password Group <i>n</i> - Lock 2 Status Register (PASS_LOCK2_PGn)..... | 2731 |
| 60.3.8 | Password Group <i>n</i> - Lock 3 Status Register (PASS_LOCK3_PGn)..... | 2732 |
| 60.4   | DCF clients.....                                                       | 2735 |
| 60.4.1 | Censorship.....                                                        | 2736 |
| 60.4.2 | Production Disable.....                                                | 2736 |
| 60.5   | Functional description.....                                            | 2738 |
| 60.5.1 | Valid password .....                                                   | 2738 |



| Section number | Title                                        | Page |
|----------------|----------------------------------------------|------|
| 60.5.2         | Censorship.....                              | 2738 |
| 60.5.3         | Debug Interface Access.....                  | 2738 |
| 60.5.4         | Flash Memory Read Protection.....            | 2739 |
| 60.5.5         | Flash Memory Write and Erase Protection..... | 2739 |
| 60.5.6         | Production Disable.....                      | 2740 |
| 60.6           | Initialization/application information.....  | 2741 |
| 60.6.1         | Reset.....                                   | 2741 |
| 60.6.2         | Setting lock bits in a password group.....   | 2741 |

## Chapter 61 Tamper Detection Module (TDM)

|         |                                                                   |      |
|---------|-------------------------------------------------------------------|------|
| 61.1    | Chip-specific TDM information.....                                | 2743 |
| 61.1.1  | Disabling the override of tamper detection.....                   | 2743 |
| 61.1.2  | TDRn_LOCKm DCF client bit mapping.....                            | 2744 |
| 61.1.3  | OTPENn DCF client bit mapping.....                                | 2747 |
| 61.2    | Overview.....                                                     | 2750 |
| 61.3    | Features.....                                                     | 2751 |
| 61.4    | External signal description.....                                  | 2751 |
| 61.5    | DCF clients.....                                                  | 2752 |
| 61.5.1  | Diary Base Address (DBA) DCF client.....                          | 2752 |
| 61.5.2  | Tamper Region Override (TO) DCF client.....                       | 2753 |
| 61.5.3  | Software Tamper Region Override Disable (STO_DIS) DCF client..... | 2754 |
| 61.5.4  | One Time Programmable Enable (OTPEN0) DCF client.....             | 2755 |
| 61.5.5  | One Time Programmable Enable (OTPEN1) DCF client.....             | 2756 |
| 61.5.6  | One Time Programmable Enable (OTPEN2) DCF client.....             | 2756 |
| 61.5.7  | One Time Programmable Enable (OTPEN3) DCF client.....             | 2757 |
| 61.5.8  | Tamper Region Assignment DCF client (TDRn_LOCK0).....             | 2758 |
| 61.5.9  | Tamper Region Assignment DCF client (TDRn_LOCK1).....             | 2759 |
| 61.5.10 | Tamper Region Assignment DCF client (TDRn_LOCK2).....             | 2760 |
| 61.5.11 | Tamper Region Assignment DCF client (TDRn_LOCK3).....             | 2760 |

| Section number | Title                                                                | Page |
|----------------|----------------------------------------------------------------------|------|
| 61.6           | Memory Map/Register Definition.....                                  | 2761 |
| 61.6.1         | TDR Status Register (TDM_TDRSR).....                                 | 2762 |
| 61.6.2         | Last Flash Programmed Address Register (TDM_LFPAR).....              | 2764 |
| 61.6.3         | Diary Base Address (TDM_DBA).....                                    | 2765 |
| 61.6.4         | Software Tamper Override Key Region (TDM_STO_KEY <sub>n</sub> )..... | 2765 |
| 61.7           | Functional description.....                                          | 2766 |
| 61.7.1         | Flash erase counter and tamper detection.....                        | 2766 |
| 61.7.2         | Assigning blocks to Tamper Detect Regions (TDR).....                 | 2768 |
| 61.7.3         | Diary.....                                                           | 2768 |
| 61.7.4         | Specifying the diary base address.....                               | 2769 |
| 61.7.5         | TDR lock status.....                                                 | 2770 |
| 61.7.6         | TDR override.....                                                    | 2770 |
| 61.7.7         | One time programmable (OTP).....                                     | 2771 |

# Chapter 1

## Preface

### 1.1 Overview

The primary objective of this document is to define the functionality of the MPC5775E Qorivva microcontroller for use by software and hardware developers. The MPC5775E is based on the Power Architecture® developed by NXP.

The information in this document is preliminary and subject to change. As with any technical documentation, it is the reader's responsibility to ensure he or she is using the most recent version of the documentation.

To locate any published errata or updates for this document, visit <http://www.nxp.com>.

### 1.2 Device versions

This release of this document describes the functionality and programming model of maskset 3N45H.

### 1.3 Audience

This manual is intended for system software and hardware developers and applications programmers who want to develop products with the MPC5775E. It is assumed that the reader understands operating systems, microprocessor system design, basic principles of software and hardware, and basic details of the Power Architecture.

### 1.4 Document organization

This document includes two major sets of chapters.

**Table 1-1. Chapter sets**

| Chapters or chapter groups                                                                                                                                                                                                     | Description                                                                        |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|
| Introduction<br>Signal Description <sup>1</sup><br>Memory Map, Interrupts, and DMA Requests<br>Device Configuration Format (DCF) Records                                                                                       | Contain information that applies to all components on the chip                     |
| Core and system modules<br>Clocking<br>Memories and memory interfaces<br>Analog measurement modules<br>Timers<br>Communication interfaces<br>Reset and boot<br>Power management<br>Debug<br>Safety modules<br>Security modules | Detail particular areas of functionality, including individual modules on the chip |

1. Details about external signal functions, their static electrical characteristics, and pad configuration settings for this chip appear in the spreadsheet that is embedded within this document: MPC5775B\_MPC5775E\_System\_IO\_Definition.xlsx. To access this spreadsheet, view the document's list of attachments.

## 1.5 Conventions

### 1.5.1 Numbering systems

The following suffixes identify different numbering systems:

| This suffix | Identifies a                                                                                                                                                       |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| b           | Binary number. For example, the binary equivalent of the number 5 is written 101b. In some cases, binary numbers are shown with the prefix 0b.                     |
| d           | Decimal number. Decimal numbers are followed by this suffix only when the possibility of confusion exists. In general, decimal numbers are shown without a suffix. |
| h           | Hexadecimal number. For example, the hexadecimal equivalent of the number 60 is written 3Ch. In some cases, hexadecimal numbers are shown with the prefix 0x.      |

### 1.5.2 Typographic notation

The following typographic notation is used throughout this document:

| Example               | Description                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>placeholder, x</i> | Items in italics are placeholders for information that you provide. Italicized text is also used for the titles of publications and for emphasis. Plain lowercase letters are also used as placeholders for single letters and numbers.                                                                                                                                                                              |
| code                  | Fixed-width type indicates text that must be typed exactly as shown. It is used for instruction mnemonics, directives, symbols, subcommands, parameters, and operators. Fixed-width type is also used for example code. Instruction mnemonics and directives in text and tables are shown in all caps; for example, BSR.                                                                                             |
| SR[SCM]               | A mnemonic in brackets represents a named field in a register. This example refers to the Scaling Mode (SCM) field in the Status Register (SR).                                                                                                                                                                                                                                                                      |
| REVNO[6:4], XAD[7:0]  | Numbers in brackets and separated by a colon represent either: <ul style="list-style-type: none"> <li>• A subset of a register's named field<br/>For example, REVNO[6:4] refers to bits 6–4 that are part of the COREREV field that occupies bits 6–0 of the REVNO register.</li> <li>• A continuous range of individual signals of a bus<br/>For example, XAD[7:0] refers to signals 7–0 of the XAD bus.</li> </ul> |

### 1.5.3 Special terms

The following terms have special meanings:

| Term       | Meaning                                                                                                                                                                                                                                                                                                                                                                                                |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| asserted   | Refers to the state of a signal as follows: <ul style="list-style-type: none"> <li>• An active-high signal is asserted when high (1).</li> <li>• An active-low signal is asserted when low (0).</li> </ul>                                                                                                                                                                                             |
| deasserted | Refers to the state of a signal as follows: <ul style="list-style-type: none"> <li>• An active-high signal is deasserted when low (0).</li> <li>• An active-low signal is deasserted when high (1).</li> </ul> <p>In some cases, deasserted signals are described as <i>negated</i>.</p>                                                                                                               |
| reserved   | Refers to a memory space, register, field, or programming setting. Writes to a reserved location can result in unpredictable functionality or behavior. <ul style="list-style-type: none"> <li>• Do not modify the default value of a reserved programming setting, such as the reset value of a reserved register field.</li> <li>• Consider undefined locations in memory to be reserved.</li> </ul> |
| w1c        | Write 1 to clear: Refers to a register bitfield that must be written as 1 to be "cleared."                                                                                                                                                                                                                                                                                                             |

## 1.6 References

In addition to this reference manual, the following documents provide information about the operation of the MPC5775E:

## References

- MPC5775E/B Microcontroller Reference Manual (document number MPC5775ERM)
- MPC5775E/MPC5775B Microcontroller Data Sheet (document number MPC5775E)
- *e200z759n3 Core Reference Manual* (document number e200z759n3CRM)
- IEEE-ISTO 5001-2003 Standard for a Global Embedded Processor Interface (Nexus)
- IEEE 1149.1-2001 standard - IEEE Standard Test Access Port and Boundary-Scan Architecture
- Power Architecture Book E V1.0, available at [www.nxp.com](http://www.nxp.com/files/32bit/doc/user_guide/BOOK_EUM.pdf): [http://www.nxp.com/files/32bit/doc/user\\_guide/BOOK\\_EUM.pdf](http://www.nxp.com/files/32bit/doc/user_guide/BOOK_EUM.pdf)

# Chapter 2

## Introduction

### 2.1 Overview

The MPC5700 family of devices is closely compatible with the MPC5600 families, while introducing new features coupled with high performance CMOS technology to provide substantial cost reduction per feature and significant performance improvement. This document summarizes features and critical performance parameters of the MPC5775E/B.

The two e200z7 host processor cores of the MPC5775E/B are compatible with the Power Architecture Book E architecture. They are 100% user-mode compatible (with floating point library) with the classic PowerPC instruction set. The Book E architecture has enhancements that improve the architecture's fit in embedded applications. In addition to the standard and VLE Power Architecture instruction sets, the cores of the MPC5775E/B have instruction support for digital signal processing (DSP).

The MPC5775E/B has two levels of memory hierarchy: separate 16 KB instruction and 16 KB data caches for each of two cores and 512 KB of on-chip SRAM as well as 4 MB of internal flash memory.

### 2.2 Device Description

#### 2.2.1 Family comparison

The following table summarizes major features of the family of devices defined in this document and compares the family with predecessor families of similar devices. The feature column represents a combination of module names and capabilities of certain modules.

Table 2-1. Family comparison

| Feature                         | MPC5676R                           | MPC5777C                                        | MPC5775B <sup>1</sup>        | MPC5775E <sup>1</sup>        |
|---------------------------------|------------------------------------|-------------------------------------------------|------------------------------|------------------------------|
| Process                         | 90 nm                              | 55 nm                                           | 55 nm                        | 55 nm                        |
| Core                            | z7 <sup>2</sup>                    | z7 <sup>2</sup>                                 | z7 <sup>2</sup>              | z7 <sup>2</sup>              |
| Number of cores                 | 2                                  | 2 + 1 lockstep core                             | 2 + 1 lockstep core          | 2 + 1 lockstep core          |
| Single Precision Floating Point | Yes                                | Yes                                             | Yes                          | Yes                          |
| SIMD                            | Yes                                | Yes                                             | Yes                          | Yes                          |
| VLE                             | Yes                                | Yes                                             | Yes                          | Yes                          |
| Cache                           | 16KB Instruction + 16KB Data       | 16KB Instruction + 16KB Data                    | 16KB Instruction + 16KB Data | 16KB Instruction + 16KB Data |
| Non-maskable interrupt          | NMI, Critical                      | NMI, Critical                                   | NMI, Critical                | NMI, Critical                |
| MMU entries                     | 32                                 | 32                                              | 32                           | 32                           |
| MMU Tool Control                | Yes                                | Yes                                             | Yes                          | Yes                          |
| MPU                             | Yes                                | Yes                                             | Yes                          | Yes                          |
| Semaphores                      | 16                                 | 16                                              | 16                           | 16                           |
| CRC channels                    | 3                                  | 3                                               | 3                            | 3                            |
| Software Watchdog Timer         | 2                                  | 2                                               | 2                            | 2                            |
| Core Nexus Class                | 3+                                 | 3+                                              | 3+                           | 3+                           |
| SRAM                            | 384KB<br>(48K Standby)             | 512KB<br>(64K Standby)                          | 512KB<br>(64K Standby)       | 512KB<br>(64K Standby)       |
| Flash memory                    | 6 MB                               | 8 MB                                            | 4 MB                         | 4 MB                         |
| Flash memory fetch accelerator  | 4 x 256 bit per core               | 16 x 256 bit per core                           | 16 x 256 bit per core        | 16 x 256 bit per core        |
| Tamper detection                | No                                 | Yes                                             | Yes                          | Yes                          |
| Security modules                | No                                 | Yes                                             | Yes                          | Yes                          |
| Safety modules                  | No                                 | Yes                                             | Yes                          | Yes                          |
| External bus                    | Yes <sup>3</sup>                   | Yes <sup>3</sup>                                | No                           | No                           |
| Calibration bus                 | 16 bit non-mux<br>16, 32 bit muxed | 16 bit non-mux<br>16, 32 bit muxed <sup>4</sup> | No                           | No                           |
| DMA channels                    | 64 + 64                            | 64 + 64                                         | 64 + 64                      | 64 + 64                      |
| DMA Nexus Class                 | 3                                  | 3                                               | 3                            | 3                            |
| Serial Interface                | 3                                  | 5                                               | 5                            | 5                            |
| FlexCAN                         | 4                                  | 4                                               | 4                            | 4                            |
| M_CAN with FD                   | No                                 | 2                                               | 2                            | 2                            |
| SPI                             | 5                                  | 5                                               | 5                            | 5                            |
| Microsecond bus downlink        | Yes                                | Yes                                             | Yes                          | Yes                          |
| SENT channels                   | No                                 | 12                                              | 12                           | 12                           |
| PSI5 channels                   | No                                 | 2                                               | 0                            | 0                            |
| Zipwire                         | No                                 | 1                                               | 0                            | 0                            |
| FlexRay                         | Yes                                | No                                              | No                           | No                           |

Table continues on the next page...



**Table 2-1. Family comparison (continued)**

| Feature                                  | MPC5676R                           | MPC5777C                           | MPC5775B <sup>1</sup>              | MPC5775E <sup>1</sup>              |
|------------------------------------------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|
| Ethernet                                 | No                                 | Yes                                | Yes                                | Yes                                |
| System Timers                            | 1 RTI, 4 PIT channels<br>4 AutoSAR | 1 RTI, 4 PIT channels<br>4 AutoSAR | 1 RTI, 4 PIT channels<br>4 AutoSAR | 1 RTI, 4 PIT channels<br>4 AutoSAR |
| eMIOS channels                           | 32                                 | 32                                 | 32                                 | 32                                 |
| eTPU channels                            | 96                                 | 96                                 | 0                                  | 96                                 |
| eTPU version                             | 3 x eTPU2                          | 3 x eTPU2                          | NA                                 | 3 x eTPU2                          |
| eTPU code memory                         | 24KB + 12KB                        | 24KB + 12KB                        | NA                                 | 24KB + 12KB                        |
| eTPU data memory                         | 6KB + 3KB                          | 6KB + 3KB                          | NA                                 | 6KB + 3KB                          |
| Interrupt controller                     | 500 sources                        | 500+ sources                       | 400+ sources                       | 400+ sources                       |
| ADC input pins                           | 64                                 | 70                                 | 40                                 | 70                                 |
| ADC input diagnostics                    | Yes                                | Yes                                | Yes                                | Yes                                |
| ADC resolution                           | 12 bit                             | 12 bit                             | 12 bit                             | 12 bit                             |
| ADC quantity                             | 4                                  | 4                                  | 2 <sup>5</sup>                     | 4                                  |
| ADC variable gain amp.                   | Yes                                | Yes                                | Yes                                | Yes                                |
| Sigma-Delta ADC                          | No                                 | 4                                  | No                                 | 4                                  |
| Reaction channels                        | No                                 | 10                                 | No                                 | No                                 |
| Temp. sensor                             | Yes                                | Yes                                | Yes                                | Yes                                |
| Decimation filters                       | 12                                 | 12                                 | No                                 | No                                 |
| Protected Port Output                    | 4                                  | 4                                  | 4                                  | 4                                  |
| Self Test Controller                     | Yes                                | Yes                                | Yes                                | Yes                                |
| Dev Tool Semaphores                      | 32                                 | 32                                 | 32                                 | 32                                 |
| PLL                                      | FM                                 | FM + non-FM                        | FM + non-FM                        | FM + non-FM                        |
| Integrated linear voltage regulator      | 3.3 V, 1.2V                        | 3.3 V, 1.2 V                       | 3.3 V, 1.2 V                       | 3.3 V, 1.2 V                       |
| Integrated switch mode voltage regulator | 1.2 V                              | 1.2 V                              | 1.2 V                              | 1.2 V                              |
| External power supplies                  | 5 V <sup>6</sup>                   | 5 V <sup>6</sup>                   | 5 V <sup>6</sup>                   | 5 V <sup>6</sup>                   |
| Low power modes                          | Stop mode<br>Slow mode             | Stop mode<br>Slow mode             | Stop mode<br>Slow mode             | Stop mode<br>Slow mode             |

1. The External Bus Interface (EBI), Zipwire (SIPI and LFAST), Decimation Filter (DECIL), PSI5 and Reaction Module (REACM) are not available on MPC5775B and MPC5775E. Any references to these modules in this Reference Manual should be ignored.
2. There are two cores on the MPC5676R, MPC5777C, MPC5775B and MPC5775E. Both cores have identical features.
3. External Bus Interface (EBI) is not available on 416 PBGA
4. Calibration bus is combined with EBI.
5. EQADCA is available on 5775B
6. External 3.3 V may be needed for external 3.3 V pins.

## 2.2.2 Block diagram

The following figure shows a top-level block diagram of the MPC5777E. The purpose of the block diagram is to show the general interconnection of functional modules through the crossbar switch.

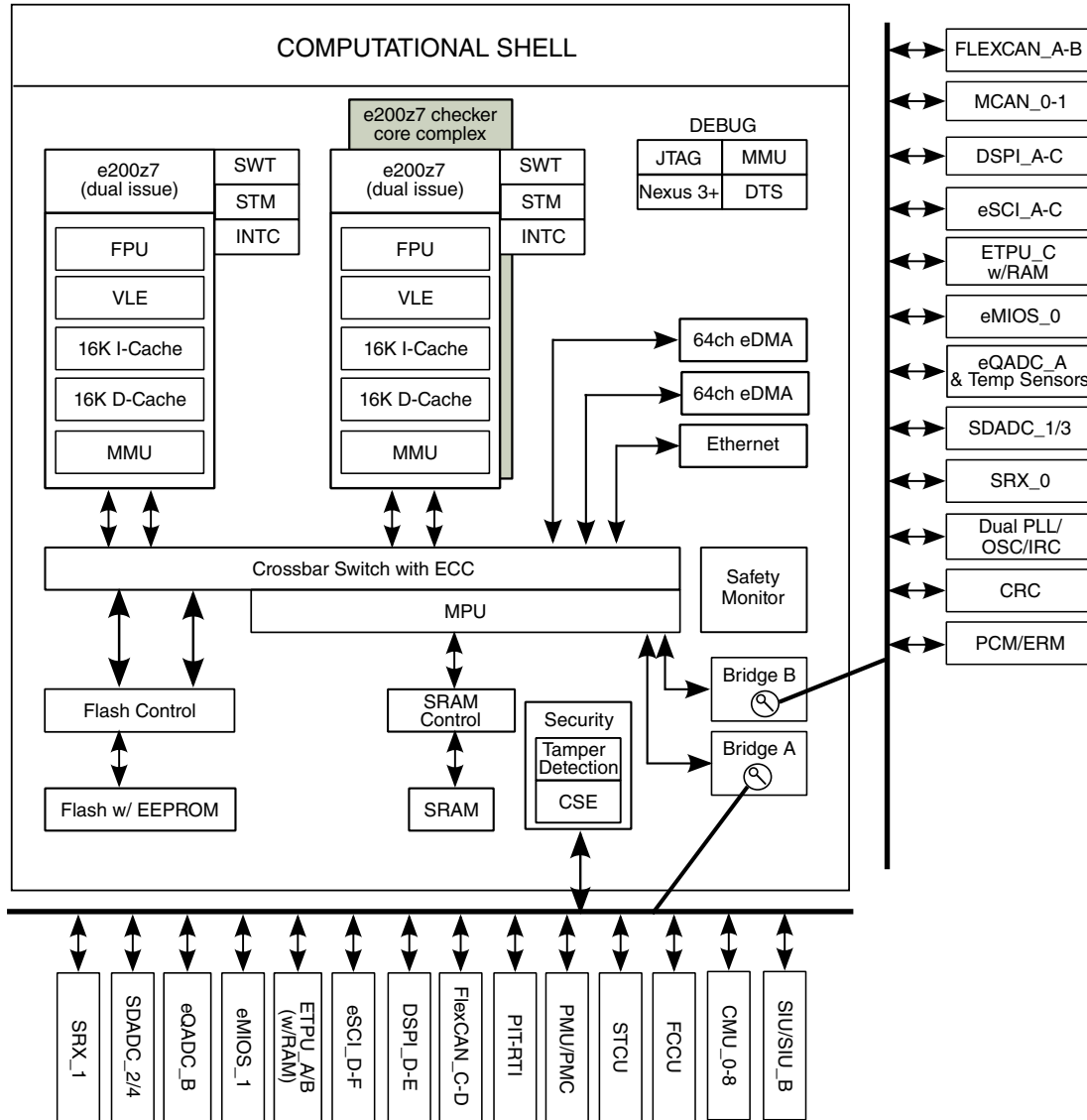


Figure 2-1. MPC5775E block diagram

## 2.2.3 Packages

The MPC5775E is offered in the following package types:

- 416-ball MAPBGA, 1 mm ball pitch, 27 mm × 27 mm outline, without EBI

## 2.2.4 Features summary

On-chip modules available within the family include the following features:

- Three dual issue, 32-bit CPU core complexes (e200z7), two of which run in lockstep
  - Power Architecture embedded specification compliance
  - Instruction set enhancement allowing variable length encoding (VLE), optional encoding of mixed 16-bit and 32-bit instructions, for code size footprint reduction
  - On the two computational cores: Signal processing extension (SPE1.1) instruction support for digital signal processing (DSP)
  - Single-precision floating point operations
  - On the two computational cores: 16 KB I-Cache and 16 KB D-Cache
  - Hardware cache coherency between cores
- 16 hardware semaphores
- 3-channel CRC module
- 4 MB on-chip flash memory
  - Supports read during program and erase operations, and multiple blocks allowing EEPROM emulation
- 512 KB on-chip general-purpose SRAM including 64 KB standby RAM
- Two multichannel direct memory access controllers (eDMA)
  - 64 channels per eDMA
- Dual core Interrupt Controller (INTC)
- Dual phase-locked loops (PLLs) with stable clock domain for peripherals and frequency modulation (FM) domain for computational shell
- Crossbar Switch architecture for concurrent access to peripherals, flash memory, or RAM from multiple bus masters with End-To-End ECC
- System Integration Unit (SIU)
- Error Injection Module (EIM) and Error Reporting Module (ERM)
- Four protected port output (PPO) pins
- Boot Assist Module (BAM) supports serial bootload via CAN or SCI
- Up to three second-generation Enhanced Time Processor Units (eTPUs)
  - 32 channels per eTPU
  - Total of 36 KB code RAM
  - Total of 9 KB parameter RAM
- Enhanced Modular Input/Output System (eMIOS) supporting 32 unified channels with each channel capable of single action, double action, pulse width modulation (PWM) and modulus counter operation
- Up to two Enhanced Queued Analog-to-Digital Converter (eQADC) modules with:

- Two separate analog converters per eQADC module
- Support for a total of 70 analog input pins, expandable to 182 inputs with off-chip multiplexers
- Up to four independent 16-bit Sigma-Delta ADCs (SDADCs)
- Ethernet (FEC)
- Two SENT Receiver (SRX) modules supporting 12 channels
- Five Deserial Serial Peripheral Interface (DSPI) modules
- Five Enhanced Serial Communication Interface (eSCI) modules
- Four Controller Area Network (FlexCAN) modules
- Two M\_CAN modules that support FD
- Fault Collection and Control Unit (FCCU)
- Clock Monitor Units (CMUs)
- Tamper Detection Module (TDM)
- Cryptographic Services Engine (CSE)
  - Complies with *Secure Hardware Extension (SHE) Functional Specification Version 1.1* security functions
  - Includes software selectable enhancement to key usage flag for MAC verification and increase in number of memory slots for security keys
- PASS module to support security features
- Nexus development interface (NDI) per IEEE-ISTO 5001-2003 standard, with some support for 2010 standard
- Device and board test support per Joint Test Action Group (JTAG) IEEE 1149.1 and 1149.7
- On-chip voltage regulator controller (VRC) that derives the core logic supply voltage from the high-voltage supply
- On-chip voltage regulator for flash memory
- Self Test capability

## Chapter 3

# Signal Description

### 3.1 Production packages

The following production packages are available for MPC5775E/B:

- BGA416

Case number and outline drawings for each package are provided in the MPC5775E data sheet.

### 3.2 Pad ring and ball maps

Refer to the MPC5775E data sheet for ball-map figures.

### 3.3 Pin function selection

#### 3.3.1 Pad output multiplexing

Most pins on the package support more than one output function. The selection of which function is driven to the pin is done by a set of Port Control Registers (SIU\_PCR) within the SIU module. These registers also configure electrical attributes such as drive strength and pull up/down selection. Refer to the SIU\_PCR section of the SIU chapter for more information.

### 3.3.2 Pin input multiplexing

Some modules in this device are allowed to receive input signals from multiple package pins. The selection of which pin drives the function is done by a set of Input Multiplexing Registers (SIU\_IMUX) within the SIU module. Refer to the SIU\_IMUX section of the SIU chapter for more information.

### 3.3.3 LVDS signal selection

A pin that is driven by a signal from a 5V pad or from an LVDS pad has a single PCR controlling both signals. Different values of PA select either the 5V signal or the LVDS signal. Note that a single LVDS pad supplies two signals: a "true" and a "complement" output. This means that one LVDS pad drives two balls, but each ball has a separate PCR (and PA field). The two LVDS signals are enabled only when the PA values in both PCRs are set to select LVDS operation. Both LVDS signals are deselected when either PA is set to a value that deselects its LVDS signal. The ball that has its PA field still set to LVDS will go to an undriven state until its PA field is updated to a non-LVDS value.

The following table is an example of the options for PA on two balls, where LVDS "true" and LVDS "complement" LVDS outputs are multiplexed with SCK\_C and SINC.

**Table 3-1. LVDS example**

| PCR235 <sup>1</sup> PA selection | PCR236 <sup>2</sup> PA selection | SCKC ball function | SINC ball function |
|----------------------------------|----------------------------------|--------------------|--------------------|
| SCKC                             | SINC                             | SCKC               | SINC               |
| LVDS                             | LVDS                             | SCK_C_LVDS_P       | SCK_C_LVDS_N       |
| SCKC                             | LVDS                             | SCKC               | undriven           |
| LVDS                             | SINC                             | undriven           | SINC               |

1. SCKC\_SCK\_C\_LVDS\_P\_GPIO235

2. SINC\_SCK\_C\_LVDS\_N\_GPIO236

[Table 3-2](#) lists all LVDS pads available on the MPC5775E/B and respective SIU PCR configuration registers and PA values. When LVDS TX functions are enabled on the balls, they can be configured to operate in mode or Micro-Second Bus mode. Also, LVDS RX pads can be configured to increase bias current and enhance timing performance. [Table 3-3](#) lists the register bits used to configure LVDS pad operating modes.

**NOTE**

Differential signal output swing can be configured on MPC5776R LVDS pads. This feature is not available on MPC5775E/B LVDS pads.

**Table 3-2. LVDS function selection**

| LVDS function | Pin polarity | Description                                       | SIU_PCR[PA] configuration              |
|---------------|--------------|---------------------------------------------------|----------------------------------------|
| SCK_C_LVDS    | +            | MSC LVDS+ downstream signal positive output clock | PCR235[PA]=10b AND<br>PCR236[PA]=10b   |
|               | –            | MSC LVDS– downstream signal negative output clock |                                        |
| SOUT_C_LVDS   | +            | MSC LVDS+ downstream signal positive output data  | PCR237[PA]=10b AND<br>PCR238[PA]=10b   |
|               | –            | MSC LVDS– downstream signal negative output data  |                                        |
| SCK_D_LVDS    | +            | MSC LVDS+ downstream signal positive output clock | PCR240[PA]=10b AND<br>PCR241[PA]=10b   |
|               | –            | MSC LVDS– downstream signal negative output clock |                                        |
| SOUT_D_LVDS   | +            | MSC LVDS+ downstream signal positive output data  | PCR242[PA]=011b AND<br>PCR243[PA]=011b |
|               | –            | MSC LVDS– downstream signal negative output data  |                                        |

**Table 3-3. LVDS pad operating modes**

| LVDS function | Configuration register/bit | Value | Pad effect            |
|---------------|----------------------------|-------|-----------------------|
| SCK_C_LVDS    | SIU_PCR235/SRC[0]          | 0     | Micro-Second Bus mode |
| SOUT_C_LVDS   | SIU_PCR237/SRC[0]          | 0     | Micro-Second Bus mode |
| SCK_D_LVDS    | SIU_PCR240/SRC[0]          | 0     | Micro-Second Bus mode |
| SOUT_D_LVDS   | SIU_PCR242/SRC[0]          | 0     | Micro-Second Bus mode |

## 3.4 External signal descriptions, multiplexing, and attributes

Details about external signal functions, their static electrical characteristics, and pad configuration settings for this chip appear in the spreadsheet that is embedded within this document:

**MPC5775B\_MPC5775E\_System\_IO\_Definition.xlsx**

To access this spreadsheet, view the document's list of attachments.

The signal properties and their electrical characteristics are set in the SIU's Pad Configuration Registers (PCRs).

### 3.4.1 IBE and OBE control special cases

The "input buffer enable" and/or "output buffer enable" of some signals are not controlled by the IBE/OBE bits of the corresponding SIU\_PCR. The following tables list these special-case signals:

- IBE always ON: [Table 3-4](#)
- OBE always ON: [Table 3-5](#)
- IBE and OBE always ON: [Table 3-6](#)
- IBE and/ or OBE controlled by a peripheral or controller: [Table 3-7](#)

**Table 3-4. IBE always ON signals**

| Signal              | Direction | PCR |
|---------------------|-----------|-----|
| BOOTCFG0            | I         | 211 |
| BOOTCFG1            | I         | 212 |
| CNRXA               | I         | 84  |
| CNRXB               | I         | 86  |
| CNRXC               | I         | 88  |
| CNRXD               | I         | 247 |
|                     |           | 194 |
| ERRORIN             | I         | 124 |
| ERROR0 <sup>1</sup> | I         | 117 |
| ERROR1 <sup>1</sup> | I         | 125 |
| ETRIG0              | I         | 244 |
| ETRIG1              | I         | 101 |
| IRQ0                | I         | 450 |
|                     |           | 193 |
| IRQ1                | I         | 451 |
|                     |           | 194 |
| IRQ10               | I         | 136 |
| IRQ11               | I         | 137 |
| IRQ12               | I         | 138 |
| IRQ13               | I         | 139 |
| IRQ14               | I         | 140 |
| IRQ15               | I         | 141 |
| IRQ2                | I         | 452 |
|                     |           | 211 |
| IRQ3                | I         | 453 |
|                     |           | 212 |
| IRQ4                | I         | 454 |
|                     |           | 208 |
| IRQ5                | I         | 455 |
|                     |           | 209 |
| IRQ6                | I         | 440 |

*Table continues on the next page...*



**Table 3-4. IBE always ON signals (continued)**

| Signal               | Direction | PCR |
|----------------------|-----------|-----|
|                      |           | 146 |
| IRQ7                 | I         | 113 |
| IRQ8                 | I         | 134 |
| IRQ9                 | I         | 135 |
| NMI                  | I         | 213 |
| PLLCFG0 <sup>2</sup> | I         | 208 |
| PLLCFG1 <sup>2</sup> | I         | 209 |
| PLLCFG2              | I         | 210 |
| RXDA                 | I         | 461 |
|                      |           | 84  |
| RXDB                 | I         | 463 |
|                      |           | 92  |
| RXDC                 | I         | 245 |
| RXDD                 | I         | 101 |
| RXDE                 | I         | 88  |
| RXDF                 | I         | 459 |
| SINA                 | I         | 94  |
| SINB                 | I         | 103 |
| SINC                 | I         | 236 |
|                      |           | 108 |
| SIND                 | I         | 472 |
|                      |           | 241 |
|                      |           | 190 |
| SINE                 | I         | 99  |
| TCRCLKA              | I         | 113 |
| TCRCLKB              | I         | 146 |
| TCRCLKC              | I         | 445 |
|                      |           | 440 |
| WKPCFG               | I         | 213 |

1. ERROR0 and ERROR1 IBE are forced ON when DCF client is set to lock ERROR0/1 functions on respective pads. See [FCCU signals](#). When function is not locked, IBE control depends on FCCU control.
2. PLLCFG0 and PLLCFG1 have no function on MPC5775E. These pin names exist only for compatibility with MPC5676R.

**Table 3-5. OBE always ON signals**

| Signal | Direction | PCR |
|--------|-----------|-----|
| CNTXA  | O         | 83  |
| CNTXB  | O         | 85  |
| CNTXC  | O         | 87  |
| CNTXD  | O         | 246 |

*Table continues on the next page...*

**Table 3-5. OBE always ON signals (continued)**

| Signal  | Direction | PCR |
|---------|-----------|-----|
|         |           | 193 |
| D_ALE   | O         | 299 |
| EMIOS14 | O         | 193 |
| EMIOS15 | O         | 194 |
| EMIOS4  | O         | 183 |
| EMIOS5  | O         | 184 |
| ETPUA0  | O         | 179 |
| ETPUA1  | O         | 180 |
| ETPUA12 | O         | 114 |
| ETPUA13 | O         | 115 |
| ETPUA14 | O         | 116 |
| ETPUA15 | O         | 117 |
| ETPUA16 | O         | 118 |
| ETPUA17 | O         | 119 |
| ETPUA18 | O         | 120 |
| ETPUA19 | O         | 121 |
| ETPUA2  | O         | 181 |
| ETPUA20 | O         | 122 |
| ETPUA21 | O         | 123 |
| ETPUA22 | O         | 124 |
| ETPUA23 | O         | 125 |
| ETPUA3  | O         | 182 |
| ETPUA4  | O         | 183 |
| ETPUA5  | O         | 184 |
| ETPUA6  | O         | 185 |
| ETPUA7  | O         | 186 |
| ETPUA8  | O         | 187 |
| ETPUA9  | O         | 188 |
| ETPUB0  | O         | 195 |
| ETPUB1  | O         | 196 |
| ETPUB16 | O         | 147 |
| ETPUB17 | O         | 148 |
| ETPUB18 | O         | 149 |
| ETPUB19 | O         | 150 |
| ETPUB2  | O         | 197 |
| ETPUB20 | O         | 151 |
| ETPUB21 | O         | 152 |
| ETPUB22 | O         | 153 |
| ETPUB23 | O         | 154 |

*Table continues on the next page...*

**Table 3-5. OBE always ON signals (continued)**

| Signal  | Direction | PCR |
|---------|-----------|-----|
| ETPUB24 | O         | 155 |
| ETPUB25 | O         | 156 |
| ETPUB26 | O         | 157 |
| ETPUB27 | O         | 158 |
| ETPUB28 | O         | 159 |
| ETPUB29 | O         | 160 |
| ETPUB3  | O         | 198 |
| ETPUB30 | O         | 161 |
| ETPUB31 | O         | 162 |
| ETPUB4  | O         | 199 |
| ETPUB5  | O         | 200 |
| ETPUB6  | O         | 201 |
| ETPUB7  | O         | 202 |
| MAA0    | O         | 464 |
| MAA1    | O         | 465 |
| MAA2    | O         | 466 |
| MAB0    | O         | 467 |
|         |           | 464 |
| MAB1    | O         | 468 |
|         |           | 465 |
| MAB2    | O         | 469 |
|         |           | 466 |
| MCKO    | O         | 219 |
| MDO0    | O         | 220 |
| PCSA1   | O         | 163 |
|         |           | 97  |
| PCSA2   | O         | 164 |
|         |           | 98  |
| PCSA3   | O         | 165 |
|         |           | 99  |
| PCSA4   | O         | 166 |
|         |           | 100 |
| PCSA5   | O         | 101 |
| PCSB1   | O         | 204 |
|         |           | 126 |
|         |           | 106 |
| PCSB2   | O         | 432 |
|         |           | 107 |
| PCSB3   | O         | 433 |

*Table continues on the next page...*

**Table 3-5. OBE always ON signals (continued)**

| Signal | Direction | PCR |
|--------|-----------|-----|
|        |           | 127 |
|        |           | 108 |
| PCSB4  | O         | 128 |
|        |           | 109 |
| PCSB5  | O         | 129 |
|        |           | 110 |
| PCSC1  | O         | 435 |
|        |           | 239 |
|        |           | 142 |
|        |           | 93  |
| PCSC2  | O         | 436 |
|        |           | 240 |
|        |           | 143 |
|        |           | 94  |
| PCSC3  | O         | 241 |
|        |           | 144 |
|        |           | 85  |
| PCSC4  | O         | 242 |
|        |           | 145 |
|        |           | 86  |
| PCSC5  | O         | 437 |
|        |           | 243 |
|        |           | 95  |
| PCSD1  | O         | 468 |
|        |           | 130 |
|        |           | 91  |
| PCSD2  | O         | 467 |
|        |           | 131 |
|        |           | 105 |
|        |           | 96  |
| PCSD3  | O         | 466 |
|        |           | 132 |
|        |           | 87  |
| PCSD4  | O         | 465 |
|        |           | 133 |
|        |           | 88  |
| PCSD5  | O         | 464 |
|        |           | 92  |
| PCSE1  | O         | 210 |

*Table continues on the next page...*

**Table 3-5. OBE always ON signals (continued)**

| Signal       | Direction | PCR |
|--------------|-----------|-----|
| PCSE2        | O         | 446 |
|              |           | 180 |
| PCSE3        | O         | 447 |
|              |           | 181 |
| PCSE4        | O         | 448 |
|              |           | 182 |
| PCSE5        | O         | 449 |
|              |           | 183 |
| RDY          | O         | 226 |
| RSTOUT       | O         | 230 |
| SCK_C_LVDS+  | O         | 235 |
| SCK_D_LVDS+  | O         | 240 |
| SOUT_C_LVDS+ | O         | 237 |
| SOUT_D_LVDS+ | O         | 242 |
| TXDD         | O         | 99  |

**Table 3-6. IBE and OBE always ON signals**

| Signal | Direction | PCR |
|--------|-----------|-----|
| TXDA   | I/O       | 460 |
|        |           | 89  |
|        |           | 83  |
| TXDB   | I/O       | 462 |
|        |           | 91  |
| TXDC   | I/O       | 244 |
| TXDD   | I/O       | 239 |
| TXDE   | I/O       | 223 |
|        |           | 100 |
| TXDE   | I/O       | 223 |
| TXDF   | I/O       | 458 |

**Table 3-7. IBE and/or OBE controlled by peripheral**

| Controlled by | Signal | Direction | IBE/OBE |
|---------------|--------|-----------|---------|
| DSPI_A        | SOUTA  | O         | OBE     |
| DSPI_B        | SOUTB  | O         | OBE     |
| DSPI_C        | SOUTC  | O         | OBE     |
| DSPI_D        | SOUTD  | O         | OBE     |
| DSPI_E        | SOUTE  | O         | OBE     |

*Table continues on the next page...*

**Table 3-7. IBE and/or OBE controlled by peripheral (continued)**

| Controlled by            | Signal   | Direction | IBE/OBE     |
|--------------------------|----------|-----------|-------------|
| Fast Ethernet Controller | FEC_MDIO | I/O       | OBE         |
| FCCU                     | ERROR0   | I/O       | IBE and OBE |
| FCCU                     | ERROR1   | I/O       | IBE and OBE |
| JTAG Controller          | TDO      | O         | OBE         |
| JTAG Controller          | TMS      | I         | OBE         |
| Nexus Controller         | EVTO     | O         | OBE         |
| Nexus Controller         | MDO1     | O         | OBE         |
| Nexus Controller         | MDO10    | O         | OBE         |
| Nexus Controller         | MDO11    | O         | OBE         |
| Nexus Controller         | MDO12    | O         | OBE         |
| Nexus Controller         | MDO13    | O         | OBE         |
| Nexus Controller         | MDO14    | O         | OBE         |
| Nexus Controller         | MDO15    | O         | OBE         |
| Nexus Controller         | MDO2     | O         | OBE         |
| Nexus Controller         | MDO3     | O         | OBE         |
| Nexus Controller         | MDO4     | O         | OBE         |
| Nexus Controller         | MDO5     | O         | OBE         |
| Nexus Controller         | MDO6     | O         | OBE         |
| Nexus Controller         | MDO7     | O         | OBE         |
| Nexus Controller         | MDO8     | O         | OBE         |
| Nexus Controller         | MDO9     | O         | OBE         |
| Nexus Controller         | MSEO0    | O         | OBE         |
| Nexus Controller         | MSEO1    | O         | OBE         |

### 3.4.2 FCCU signals

A DCF client, as defined in the DCF Records chapter, is used to enable ERROR0, ERROR1, and ERRORIN functions on each pin independently. If one of the ERROR functions is disabled by the DCF client at power-on reset (POR), the associated pin resets as GPIO.

If the ERROR0, ERROR1, or ERRORIN function is enabled on the pin by the DCF client bits E0E, E1E, or EINE, respectively, the ERROR function and its attributes are locked on the pin until the next POR. A reset of the device that is not a POR will not disrupt the ERROR pins' function or state.

When the ERROR function is enabled by the DCF client at POR, the following applies to the ERROR0 and ERROR1 pins:

- POR state of the pins is Hi-Z
- Input to the FCCU is enabled
- FCCU drives the output data on the pin and no muxing change can occur on the pin, but software can read the pin state via GPIO if it enables the input buffer in the corresponding PCR
- Bits in the DCF client select the open-drain or push-pull output configuration (see the DCF Records chapter for details):
  - E1OD (ERROR1)
  - E0OD (ERROR0)
- The slew rate of the ERROR0/1 pins is set to the maximum
- Weak pull up/down is disabled, which overrides the weak pull up/down configuration for the GPIO pins determined by the WKPCFG pin or WKPCFG DCF client

When the ERROR function is enabled by the DCF client at POR, the following applies to the ERRORIN pin:

- POR state of the pins is Hi-Z
- Input to the FCCU is enabled
- Weak pull up/down is disabled, which overrides the weak pull up/down configuration for the GPIO pins determined by the WKPCFG pin or WKPCFG DCF client





# Chapter 4

## Memory Map, Interrupts, and DMA Requests

### 4.1 Memory map

All addresses in the device are identified in the following table. The addresses represent the physical addresses assigned to each block. Logical addresses are translated by the MMU into physical addresses.

Under software control of the Memory Management Unit (MMU), the logical addresses allocated to IP blocks may be changed on a minimum of a 1 KB boundary.

#### NOTE

Address ranges not declared in the memory map table must be considered reserved.

**Table 4-1. Memory map**

| Address range                                          | Module                                                             |
|--------------------------------------------------------|--------------------------------------------------------------------|
| Flash memory—see <a href="#">Table 4-2</a> for details |                                                                    |
| 0x0000_0000–0x0003_FFFF                                | Flash memory Low/Mid Blocks (256K)                                 |
| 0x0004_0000–0x003F_FFFF                                | Reserved                                                           |
| 0x0040_0000–0x0040_3FFF                                | Flash memory UTEST (16K)—see <a href="#">Table 4-3</a> for details |
| 0x0040_4000–0x005F_FFFF                                | Reserved                                                           |
| 0x0060_0000–0x0060_7FFF                                | Flash memory High Blocks (32K)                                     |
| 0x0060_8000–0x007F_FFFF                                | Reserved                                                           |
| 0x0080_0000–0x00BF_FFFF                                | Flash memory Large Blocks (4M)                                     |
| 0x00C0_0000–0x00FF_FFFF                                | Reserved                                                           |
| 0x0100_0000–0x0FFF_FFFF                                | Reserved                                                           |
| 0x1000_0000–0x1FFF_FFFF                                | Emulation remapping of flash memory using MMU                      |
| External memory                                        |                                                                    |
| 0x2000_0000–0x3FFF_FFFF                                | Reserved                                                           |
| System RAM                                             |                                                                    |
| 0x4000_0000–0x4000_FFFF                                | Standby SRAM on XBAR Slave Port 2 (64 KB)                          |

*Table continues on the next page...*

Table 4-1. Memory map (continued)

| Address range                | Module                             |
|------------------------------|------------------------------------|
| 0x4001_0000–0x4003_FFFF      | SRAM on XBAR Slave Port 2 (192 KB) |
| 0x4004_0000–0x4007_FFFF      | SRAM on XBAR Slave Port 4 (256 KB) |
| 0x4008_0000–0xC3DF_FFFF      | Reserved                           |
| Peripheral Bridge A (AIPS_1) |                                    |
| 0xC3E0_0000–0xC3E0_3FFF      | Reserved                           |
| 0xC3E0_4000–0xC3E0_7FFF      | Input Glitch Filter 1              |
| 0xC3E0_8000–0xC3E1_3FFF      | Reserved                           |
| 0xC3E1_4000–0xC3E1_7FFF      | Reserved                           |
| 0xC3E1_8000–0xC3E4_3FFF      | Reserved                           |
| 0xC3E4_4000–0xC3E4_7FFF      | SDADC_2                            |
| 0xC3E4_8000–0xC3E4_BFFF      | SDADC_4                            |
| 0xC3E4_C000–0xC3E5_3FFF      | Reserved                           |
| 0xC3E5_4000–0xC3E5_7FFF      | eQADC_B                            |
| 0xC3E5_8000–0xC3E5_BFFF      | Reserved                           |
| 0xC3E5_C000–0xC3E5_FFFF      | SRX_1                              |
| 0xC3E6_0000–0xC3E6_3FFF      | FlexCAN_C                          |
| 0xC3E6_4000–0xC3E6_7FFF      | FlexCAN_D                          |
| 0xC3E6_8000–0xC3E6_C1FF      | Reserved                           |
| 0xC3E6_C200–0xC3E6_C23F      | CMU_0                              |
| 0xC3E6_C240–0xC3E6_C27F      | CMU_1                              |
| 0xC3E6_C280–0xC3E6_C2BF      | CMU_2                              |
| 0xC3E6_C2C0–0xC3E6_C2FF      | CMU_3                              |
| 0xC3E6_C300–0xC3E6_C33F      | CMU_4                              |
| 0xC3E6_C340–0xC3E6_C37F      | CMU_5                              |
| 0xC3E6_C380–0xC3E6_C3BF      | CMU_6                              |
| 0xC3E6_C3C0–0xC3E6_C3FF      | CMU_7                              |
| 0xC3E6_C400–0xC3E6_C43F      | CMU_8                              |
| 0xC3E7_C440–0xC3EF_FFFF      | Not allocated                      |
| 0xC3F0_0000–0xC3F0_3FFF      | PBRIDGE registers                  |
| 0xC3F0_4000–0xC3F7_FFFF      | Reserved <sup>1</sup>              |
| 0xC3F8_0000–0xC3F8_3FFF      | FCCU                               |
| 0xC3F8_4000–0xC3F8_7FFF      | Reserved                           |
| 0xC3F8_8000–0xC3F8_FFFF      | Reserved                           |
| 0xC3F9_0000–0xC3F9_3FFF      | SIU                                |
| 0xC3F9_4000–0xC3F9_7FFF      | Reserved                           |
| 0xC3F9_8000–0xC3F9_BFFF      | SIU_B                              |
| 0xC3F9_C000–0xC3F9_FFFF      | Reserved                           |
| 0xC3FA_0000–0xC3FA_3FFF      | eMIOS_1                            |
| 0xC3FA_4000–0xC3FA_FFFF      | Reserved                           |

Table continues on the next page...

**Table 4-1. Memory map (continued)**

| Address range                | Module                      |
|------------------------------|-----------------------------|
| 0xC3FB_0000–0xC3FB_3FFF      | eSCI_D                      |
| 0xC3FB_4000–0xC3FB_7FFF      | eSCI_E                      |
| 0xC3FB_8000–0xC3FB_BFFF      | eSCI_F                      |
| 0xC3FB_C000–0xC3FB_FFFF      | PMC                         |
| 0xC3FC_0000–0xC3FC_3FFF      | eTPU_A/B                    |
| 0xC3FC_4000–0xC3FC_7FFF      | Reserved                    |
| 0xC3FC_8000–0xC3FC_BFFF      | eTPU Parameter RAM          |
| 0xC3FC_C000–0xC3FC_FFFF      | eTPU Parameter RAM Mirror   |
| 0xC3FD_0000–0xC3FD_3FFF      | eTPU Code RAM               |
| 0xC3FD_4000–0xC3FD_7FFF      | eTPU Code RAM               |
| 0xC3FD_8000–0xC3FD_BFFF      | Reserved                    |
| 0xC3FD_C000–0xC3FD_FFFF      | DSPI_D                      |
| 0xC3FE_0000–0xC3FE_3FFF      | DSPI_E                      |
| 0xC3FE_4000–0xC3FE_FFFF      | Reserved                    |
| 0xC3FF_0000–0xC3FF_3FFF      | PIT_RTI                     |
| 0xC3FF_4000–0xC3FF_7FFF      | STCU                        |
| 0xC3FF_8000–0xFFDF_FFFF      | Reserved                    |
| Peripheral Bridge B (AIPS_0) |                             |
| 0xFFE0_0000–0xFFE0_3FFF      | eTPU_C                      |
| 0xFFE0_4000–0xFFE0_7FFF      | Reserved                    |
| 0xFFE0_8000–0xFFE0_BFFF      | eTPU_C Parameter RAM        |
| 0xFFE0_C000–0xFFE0_FFFF      | eTPU_C Parameter RAM Mirror |
| 0xFFE1_0000–0xFFE1_3FFF      | eTPU_C Code RAM             |
| 0xFFE1_4000–0xFFE1_FFFF      | Reserved                    |
| 0xFFE2_0000–0xFFE2_3FFF      | Input Glitch Filter 0       |
| 0xFFE2_4000–0xFFE2_FFFF      | Reserved                    |
| 0xFFE3_0000–0xFFE3_3FFF      | M_CAN_0                     |
| 0xFFE3_4000–0xFFE3_7FFF      | M_CAN RAM                   |
| 0xFFE3_8000–0xFFE3_BFFF      | M_CAN_1                     |
| 0xFFE3_C000–0xFFE3_FFFF      | Reserved                    |
| 0xFFE4_0000–0xFFE4_3FFF      | Reserved                    |
| 0xFFE4_4000–0xFFE4_7FFF      | SDADC_1                     |
| 0xFFE4_8000–0xFFE4_BFFF      | SDADC_3                     |
| 0xFFE4_C000–0xFFE5_BFFF      | Reserved                    |
| 0xFFE5_C000–0xFFE5_FFFF      | SRX_0                       |
| 0xFFE6_0000–0xFFE6_3FFF      | Reserved                    |
| 0xFFE6_4000–0xFFE6_7FFF      | eMIOS_0                     |
| 0xFFE6_8000–0xFFE6_BFFF      | CRC                         |
| 0xFFE6_C000–0xFFE6_FFFF      | Reserved                    |

*Table continues on the next page...*

Table 4-1. Memory map (continued)

| Address range             | Module                     |
|---------------------------|----------------------------|
| 0xFFE7_0000–0xFFE7_3FFF   | PCS                        |
| 0xFFE7_4000–0xFFE7_7FFF   | PLLDIG                     |
| 0xFFE7_8000–0xFFE7_BFFF   | DTS                        |
| 0xFFE7_C000–0xFFE7_FFFF   | JDC                        |
| 0xFFE8_0000–0xFFEF_FFFF   | Not allocated <sup>2</sup> |
| 0xFFFF0_0000–0xFFFF0_3FFF | PBRIDGE registers          |
| 0xFFFF0_4000–0xFFFF0_7FFF | XBAR                       |
| 0xFFFF0_8000–0xFFFF0_BFFF | XBIC                       |
| 0xFFFF0_C000–0xFFFF0_FFFF | MPU_0                      |
| 0xFFFF1_0000–0xFFFF1_3FFF | MPU_1                      |
| 0xFFFF1_4000–0xFFFF1_7FFF | CSE                        |
| 0xFFFF1_8000–0xFFFF1_BFFF | TDM                        |
| 0xFFFF1_C000–0xFFFF1_FFFF | PRAMC_0                    |
| 0xFFFF2_0000–0xFFFF2_3FFF | PRAMC_1                    |
| 0xFFFF2_4000–0xFFFF2_7FFF | SEMA4                      |
| 0xFFFF2_8000–0xFFFF3_3FFF | Reserved <sup>1</sup>      |
| 0xFFFF3_4000–0xFFFF3_7FFF | SWT_B                      |
| 0xFFFF3_8000–0xFFFF3_BFFF | SWT_A                      |
| 0xFFFF3_C000–0xFFFF3_FFFF | STM_A                      |
| 0xFFFF4_0000–0xFFFF4_3FFF | Reserved <sup>1</sup>      |
| 0xFFFF4_4000–0xFFFF4_7FFF | eDMA_A                     |
| 0xFFFF4_8000–0xFFFF4_BFFF | INTC                       |
| 0xFFFF4_C000–0xFFFF4_FFFF | FEC                        |
| 0xFFFF5_0000–0xFFFF5_3FFF | PCU                        |
| 0xFFFF5_4000–0xFFFF5_7FFF | eDMA_B                     |
| 0xFFFF5_8000–0xFFFF5_FFFF | Reserved <sup>1</sup>      |
| 0xFFFF6_0000–0xFFFF6_3FFF | ERM                        |
| 0xFFFF6_4000–0xFFFF6_7FFF | EIM                        |
| 0xFFFF6_8000–0xFFFF6_BFFF | PFLASH Configuration       |
| 0xFFFF6_C000–0xFFFF6_C1FF | PCM                        |
| 0xFFFF6_C200–0xFFFF7_FFFF | Reserved <sup>1</sup>      |
| 0xFFFF8_0000–0xFFFF8_3FFF | eQADC_A                    |
| 0xFFFF8_4000–0xFFFF8_7FFF | Reserved                   |
| 0xFFFF8_8000–0xFFFF8_87FF | Reserved                   |
| 0xFFFF8_8800–0xFFFF8_8FFF | Reserved                   |
| 0xFFFF8_9000–0xFFFF8_97FF | Reserved                   |
| 0xFFFF8_9800–0xFFFF8_9FFF | Reserved                   |
| 0xFFFF8_A000–0xFFFF8_A7FF | Reserved                   |
| 0xFFFF8_A800–0xFFFF8_AFFF | Reserved                   |

Table continues on the next page...

**Table 4-1. Memory map (continued)**

| Address range             | Module                     |
|---------------------------|----------------------------|
| 0xFFFF8_B000–0xFFFF8_B7FF | Reserved                   |
| 0xFFFF8_B800–0xFFFF8_BFFF | Reserved                   |
| 0xFFFF8_C000–0xFFFF8_C7FF | Reserved                   |
| 0xFFFF8_C800–0xFFFF8_CFFF | Reserved                   |
| 0xFFFF8_D000–0xFFFF8_D7FF | Reserved                   |
| 0xFFFF8_D800–0xFFFF8_DFFF | Reserved                   |
| 0xFFFF8_E000–0xFFFF8_FFFF | Not used                   |
| 0xFFFF9_0000–0xFFFF9_3FFF | DSPI_A                     |
| 0xFFFF9_4000–0xFFFF9_7FFF | DSPI_B                     |
| 0xFFFF9_8000–0xFFFF9_BFFF | DSPI_C                     |
| 0xFFFF9_C000–0xFFFFA_FFFF | Reserved                   |
| 0xFFFFB_0000–0xFFFFB_3FFF | eSCI_A                     |
| 0xFFFFB_4000–0xFFFFB_7FFF | eSCI_B                     |
| 0xFFFFB_8000–0xFFFFB_BFFF | eSCI_C                     |
| 0xFFFFB_C000–0xFFFFB_FFFF | Reserved                   |
| 0xFFFFC_0000–0xFFFFC_3FFF | FlexCAN_A                  |
| 0xFFFFC_4000–0xFFFFC_7FFF | FlexCAN_B                  |
| 0xFFFFC_8000–0xFFFFD_7FFF | Reserved                   |
| 0xFFFFD_8000–0xFFFFD_BFFF | Reserved                   |
| 0xFFFFD_C000–0xFFFFE_3FFF | Reserved                   |
| 0xFFFFE_4000–0xFFFFE_7FFF | Reserved                   |
| 0xFFFFE_8000–0xFFFFE_BFFF | C55FMC flash memory module |
| 0xFFFFE_C000–0xFFFFE_FFFF | Reserved                   |
| 0xFFFFF_0000–0xFFFFF_3FFF | ATX                        |
| 0xFFFFF_4000–0xFFFFF_7FFF | PASS                       |
| 0xFFFFF_8000–0xFFFFF_BFFF | SSCM                       |
| 0xFFFFF_C000–0xFFFFF_FFFF | BAM                        |

1. Any attempted access to this reserved space, which is part of the memory map for on-platform peripherals, generates a transfer error.
2. Any attempted access to this unallocated space generates a transfer error.

**Table 4-2. Flash memory map**

| Start address | End address | Allocated size (KB) | RWW partition | Block number | Block type    | Example use |
|---------------|-------------|---------------------|---------------|--------------|---------------|-------------|
| 0x00000000    | 0x0000FFFF  | 64                  | 0             | 0            | Low           | EEPROM data |
| 0x00010000    | 0x0001FFFF  | 64                  | 1             | 1            | Low           | EEPROM data |
| 0x00020000    | 0x0002FFFF  | 64                  | 2             | 0            | Mid           | EEPROM data |
| 0x00030000    | 0x0003FFFF  | 64                  | 3             | 1            | Mid           | EEPROM data |
| 0x00040000    | 0x003FFFFF  | 3840                | —             | —            | Not available | n/a         |

*Table continues on the next page...*

Table 4-2. Flash memory map (continued)

| Start address | End address  | Allocated size (KB) | RWW partition | Block number | Block type    | Example use                   |
|---------------|--------------|---------------------|---------------|--------------|---------------|-------------------------------|
| 0x00400000    | 0x00403FFF   | 16                  | 0             | 0            | UTEST         | See <a href="#">Table 4-3</a> |
| 0x00404000    | 0x005FFFFFFF | 2032                | —             | —            | Not available | n/a                           |
| 0x00600000    | 0x00603FFF   | 16                  | 4             | 0            | High          | CSE                           |
| 0x00604000    | 0x00607FFF   | 16                  | 5             | 1            | High          | CSE                           |
| 0x00608000    | 0x007FFFFFFF | 2016                | —             | —            | Not available | n/a                           |
| 0x00800000    | 0x0083FFFF   | 256                 | 6             | 0            | Large         | Boot                          |
| 0x00840000    | 0x0087FFFF   | 256                 | 6             | 1            | Large         | Application code              |
| 0x00880000    | 0x008BFFFF   | 256                 | 6             | 2            | Large         | Application code              |
| 0x008C0000    | 0x008FFFFFFF | 256                 | 6             | 3            | Large         | Application code              |
| 0x00900000    | 0x0093FFFF   | 256                 | 6             | 4            | Large         | Application code              |
| 0x00940000    | 0x0097FFFF   | 256                 | 6             | 5            | Large         | Application code              |
| 0x00980000    | 0x009BFFFF   | 256                 | 6             | 6            | Large         | Application code              |
| 0x009C0000    | 0x009FFFFFFF | 256                 | 6             | 7            | Large         | Application code              |
| 0x00A00000    | 0x00A3FFFF   | 256                 | 7             | 8            | Large         | Application code              |
| 0x00A40000    | 0x00A7FFFF   | 256                 | 7             | 9            | Large         | Application code              |
| 0x00A80000    | 0x00ABFFFF   | 256                 | 7             | 10           | Large         | Application code              |
| 0x00AC0000    | 0x00AFFFFFFF | 256                 | 7             | 11           | Large         | Application code              |
| 0x00B00000    | 0x00B3FFFF   | 256                 | 7             | 12           | Large         | Application code              |
| 0x00B40000    | 0x00B7FFFF   | 256                 | 7             | 13           | Large         | Application code              |
| 0x00B80000    | 0x00BBFFFF   | 256                 | 7             | 14           | Large         | Application code              |
| 0x00BC0000    | 0x00BFFFFFFF | 256                 | 7             | 15           | Large         | Application code              |
| 0x00C00000    | 0x00C3FFFF   | 256                 | —             | 16           | —             | Reserved                      |
| 0x00C40000    | 0x00C7FFFF   | 256                 | —             | 17           | —             | Reserved                      |
| 0x00C80000    | 0x00CBFFFF   | 256                 | —             | 18           | —             | Reserved                      |
| 0x00CC0000    | 0x00CFFFFFFF | 256                 | —             | 19           | —             | Reserved                      |
| 0x00D00000    | 0x00D3FFFF   | 256                 | —             | 20           | —             | Reserved                      |
| 0x00D40000    | 0x00D7FFFF   | 256                 | —             | 21           | —             | Reserved                      |
| 0x00D80000    | 0x00DBFFFF   | 256                 | —             | 22           | —             | Reserved                      |
| 0x00DC0000    | 0x00DFFFFFFF | 256                 | —             | 23           | —             | Reserved                      |
| 0x00E00000    | 0x00E3FFFF   | 256                 | —             | 24           | —             | Reserved                      |
| 0x00E40000    | 0x00E7FFFF   | 256                 | —             | 25           | —             | Reserved                      |
| 0x00E80000    | 0x00EBFFFF   | 256                 | —             | 26           | —             | Reserved                      |
| 0x00EC0000    | 0x00EFFFFFFF | 256                 | —             | 27           | —             | Reserved                      |
| 0x00F00000    | 0x00F3FFFF   | 256                 | —             | 28           | —             | Reserved                      |
| 0x00F40000    | 0x00F7FFFF   | 256                 | —             | 29           | —             | Reserved                      |
| 0x00F80000    | 0x00FBFFFF   | 256                 | —             | 30           | —             | Reserved                      |
| 0x00FC0000    | 0x00FFFFFFF  | 256                 | —             | 31           | —             | Reserved                      |

**Table 4-3. UTEST flash memory map**

| Start address | End address | Size (bytes) | Description                            | Notes |
|---------------|-------------|--------------|----------------------------------------|-------|
| 0x00400000    | 0x00400001  | 2            | Temp Sensor 0 Calibration A            | 1     |
| 0x00400002    | 0x00400003  | 2            | Temp Sensor 0 Calibration B            | 1     |
| 0x00400004    | 0x00400005  | 2            | Temp Sensor 1 Calibration A            | 1     |
| 0x00400006    | 0x00400007  | 2            | Temp Sensor 1 Calibration B            | 1     |
| 0x00400008    | 0x0040000B  | 4            | Reserved                               | —     |
| 0x0040000C    | 0x0040000F  | 4            | Test Mode Disable Seal                 | 2     |
| 0x00400010    | 0x0040001F  | 16           | Test Mode Disable Block Select Group A | 3     |
| 0x00400020    | 0x0040002F  | 16           | Reserved                               | —     |
| 0x00400030    | 0x0040003F  | 16           | Test Mode Disable Block Select Group B | 4     |
| 0x00400040    | 0x0040005F  | 32           | Customer Single Bit Correction Area    | 5     |
| 0x00400060    | 0x0040007F  | 32           | Customer Double Bit Detection Area     | 5     |
| 0x00400080    | 0x0040009F  | 32           | Customer EDC after ECC Area            | 5     |
| 0x004000A0    | 0x004000BF  | 32           | Unique Chip Identifier (UID)           | 6     |
| 0x004000C0    | 0x004000C3  | 4            | Reserved                               | —     |
| 0x004000C4    | 0x004000C7  | 4            | Reserved                               | —     |
| 0x004000C8    | 0x004000FF  | 56           | Reserved                               | —     |
| 0x00400100    | 0x00400103  | 4            | Test Mode Disable Override Passcode    | 7     |
| 0x00400104    | 0x0040011F  | 28           | Reserved                               | 7     |
| 0x00400120    | 0x0040013F  | 32           | JTAG Password                          | 7     |
| 0x00400140    | 0x0040015F  | 32           | PASS Password Group 0                  | 7     |
| 0x00400160    | 0x0040017F  | 32           | PASS Password Group 1                  | 7     |
| 0x00400180    | 0x0040019F  | 32           | PASS Password Group 2                  | 7     |
| 0x004001A0    | 0x004001BF  | 32           | PASS Password Group 3                  | 7     |
| 0x004001C0    | 0x004001DF  | 32           | Reserved                               | 7     |
| 0x004001E0    | 0x004001FF  | 32           | Reserved                               | 7     |
| 0x00400200    | 0x0040020F  | 16           | Life cycle slot 0 - MCU Production     | 8     |
| 0x00400210    | 0x0040021F  | 16           | Life cycle slot 1 - Customer Delivery  | 9     |
| 0x00400220    | 0x0040022F  | 16           | Life cycle slot 2 - OEM Production     | 10    |
| 0x00400230    | 0x0040023F  | 16           | Life cycle slot 3 - In Field           | 11    |
| 0x00400240    | 0x0040024F  | 16           | Life cycle slot 4 - Failure Analysis   | 12    |
| 0x00400250    | 0x004002FF  | 176          | Reserved                               | —     |
| 0x00400300    | 0x00400307  | 8            | DCF start record                       | 13    |
| 0x00400308    | 0x00400FFF  | 3320         | DCF records                            | 14    |
| 0x00401000    | 0x00403FFF  | 12288        | Reserved for customer OTP data         | 15    |

1. Programmed by NXP during test. Contains the temperature sensor and bandgap ADC readings to be used by software to improve the accuracy of the temp sensor readings.
2. Programmed by NXP during test. In production devices, UTEST is sealed (one-time programmable) from the factory.
3. Programmed by the customer to select which blocks cannot be tested in Failure Analysis.
4. Secondary location that can be programmed by the customer to select which blocks cannot be tested in Failure Analysis.
5. Programmed by NXP to include ECC/EDC errors to allow testing of ECC/EDC hardware.

## Interrupt vector assignments

6. Programmed by NXP during test. The UID includes information on wafer lot, X/Y-position on the wafer, manufacturing data, test results, and company ID (65 for NXP).
7. Protected from read operations by flash memory BIU from OEM Production life cycle. Programmed by customer. Becomes unreadable (hidden) as soon as customer programs life cycle to OEM Production or later.
8. MCU Production life cycle slot. Programmed by NXP during test.
9. Customer Delivery life cycle slot. Programmed by NXP during test.
10. OEM Production life cycle slot. Programmed by customer during ECO module production/test.
11. In Field life cycle slot. Programmed by customer during ECO module production/test.
12. Failure Analysis life cycle slot. Programmed by the customer on returned modules.
13. DCF start record. Programmed by NXP during test. Contiguous list of DCF records start at this address.
14. Contiguous list of DCF records. Initial records programmed by NXP. Subsequent records added to the end of the list by the customer.
15. Programmed by the customer.

## 4.2 Interrupt vector assignments

### NOTE

The following table shows the mapping between interrupt sources and interrupt vectors. Some interrupt sources have a separate mechanism for enabling the interrupt, such as an interrupt enable register field. To support the mapping between a source and a vector, any such separate mechanism must be set to enable the interrupt source.

**Table 4-4. Interrupt vectors and request sources**

| Offset | Vector | Source                                                   | Module     | Description                                                    |
|--------|--------|----------------------------------------------------------|------------|----------------------------------------------------------------|
| 0x0000 | 0      | INTC_SSCIR0[CLR0]                                        | Software   | INTC software settable Clear flag 0                            |
| 0x0010 | 1      | INTC_SSCIR1[CLR1]                                        | Software   | INTC software settable Clear flag 1                            |
| 0x0020 | 2      | INTC_SSCIR2[CLR2]                                        | Software   | INTC software settable Clear flag 2                            |
| 0x0030 | 3      | INTC_SSCIR3[CLR3]                                        | Software   | INTC software settable Clear flag 3                            |
| 0x0040 | 4      | INTC_SSCIR4[CLR4]                                        | Software   | INTC software settable Clear flag 4                            |
| 0x0050 | 5      | INTC_SSCIR5[CLR5]                                        | Software   | INTC software settable Clear flag 5                            |
| 0x0060 | 6      | INTC_SSCIR6[CLR6]                                        | Software   | INTC software settable Clear flag 6                            |
| 0x0070 | 7      | INTC_SSCIR7[CLR7]                                        | Software   | INTC software settable Clear flag 7                            |
| 0x0080 | 8      | SWTA_IR[TIF]                                             | Watchdog A | Software Watchdog A Interrupt flag                             |
| 0x0090 | 9      | FCCU_IRQ_STAT[ALRM_STAT]  <br>FCCU_IRQ_STAT[CFG_TO_STAT] | FCCU       | FCCU ALARM state entry   FCCU CONFIG<br>state watchdog timeout |
| 0x00A0 | 10     | EDMA_ERL[ERR31:ERR0]                                     | eDMA_A     | eDMA channel Error flags 0-31                                  |
| 0x00B0 | 11     | EDMA_IRQRL[INT00]                                        | eDMA_A     | eDMA channel Interrupt 0                                       |
| 0x00C0 | 12     | EDMA_IRQRL[INT01]                                        | eDMA_A     | eDMA channel Interrupt 1                                       |
| 0x00D0 | 13     | EDMA_IRQRL[INT02]                                        | eDMA_A     | eDMA channel Interrupt 2                                       |
| 0x00E0 | 14     | EDMA_IRQRL[INT03]                                        | eDMA_A     | eDMA channel Interrupt 3                                       |
| 0x00F0 | 15     | EDMA_IRQRL[INT04]                                        | eDMA_A     | eDMA channel Interrupt 4                                       |
| 0x0100 | 16     | EDMA_IRQRL[INT05]                                        | eDMA_A     | eDMA channel Interrupt 5                                       |

*Table continues on the next page...*



**Table 4-4. Interrupt vectors and request sources (continued)**

| Offset | Vector | Source                        | Module  | Description                                                                     |
|--------|--------|-------------------------------|---------|---------------------------------------------------------------------------------|
| 0x0110 | 17     | EDMA_IRQRL[INT06]             | eDMA_A  | eDMA channel Interrupt 6                                                        |
| 0x0120 | 18     | EDMA_IRQRL[INT07]             | eDMA_A  | eDMA channel Interrupt 7                                                        |
| 0x0130 | 19     | EDMA_IRQRL[INT08]             | eDMA_A  | eDMA channel Interrupt 8                                                        |
| 0x0140 | 20     | EDMA_IRQRL[INT09]             | eDMA_A  | eDMA channel Interrupt 9                                                        |
| 0x0150 | 21     | EDMA_IRQRL[INT10]             | eDMA_A  | eDMA channel Interrupt 10                                                       |
| 0x0160 | 22     | EDMA_IRQRL[INT11]             | eDMA_A  | eDMA channel Interrupt 11                                                       |
| 0x0170 | 23     | EDMA_IRQRL[INT12]             | eDMA_A  | eDMA channel Interrupt 12                                                       |
| 0x0180 | 24     | EDMA_IRQRL[INT13]             | eDMA_A  | eDMA channel Interrupt 13                                                       |
| 0x0190 | 25     | EDMA_IRQRL[INT14]             | eDMA_A  | eDMA channel Interrupt 14                                                       |
| 0x01A0 | 26     | EDMA_IRQRL[INT15]             | eDMA_A  | eDMA channel Interrupt 15                                                       |
| 0x01B0 | 27     | EDMA_IRQRL[INT16]             | eDMA_A  | eDMA channel Interrupt 16                                                       |
| 0x01C0 | 28     | EDMA_IRQRL[INT17]             | eDMA_A  | eDMA channel Interrupt 17                                                       |
| 0x01D0 | 29     | EDMA_IRQRL[INT18]             | eDMA_A  | eDMA channel Interrupt 18                                                       |
| 0x01E0 | 30     | EDMA_IRQRL[INT19]             | eDMA_A  | eDMA channel Interrupt 19                                                       |
| 0x01F0 | 31     | EDMA_IRQRL[INT20]             | eDMA_A  | eDMA channel Interrupt 20                                                       |
| 0x0200 | 32     | EDMA_IRQRL[INT21]             | eDMA_A  | eDMA channel Interrupt 21                                                       |
| 0x0210 | 33     | EDMA_IRQRL[INT22]             | eDMA_A  | eDMA channel Interrupt 22                                                       |
| 0x0220 | 34     | EDMA_IRQRL[INT23]             | eDMA_A  | eDMA channel Interrupt 23                                                       |
| 0x0230 | 35     | EDMA_IRQRL[INT24]             | eDMA_A  | eDMA channel Interrupt 24                                                       |
| 0x0240 | 36     | EDMA_IRQRL[INT25]             | eDMA_A  | eDMA channel Interrupt 25                                                       |
| 0x0250 | 37     | EDMA_IRQRL[INT26]             | eDMA_A  | eDMA channel Interrupt 26                                                       |
| 0x0260 | 38     | EDMA_IRQRL[INT27]             | eDMA_A  | eDMA channel Interrupt 27                                                       |
| 0x0270 | 39     | EDMA_IRQRL[INT28]             | eDMA_A  | eDMA channel Interrupt 28                                                       |
| 0x0280 | 40     | EDMA_IRQRL[INT29]             | eDMA_A  | eDMA channel Interrupt 29                                                       |
| 0x0290 | 41     | EDMA_IRQRL[INT30]             | eDMA_A  | eDMA channel Interrupt 30                                                       |
| 0x02A0 | 42     | EDMA_IRQRL[INT31]             | eDMA_A  | eDMA channel Interrupt 31                                                       |
| 0x02B0 | 43     | SIU_PCSIFR[PCSI]              | PCS     | Progressive Clock Switch Interrupt                                              |
| 0x02C0 | 44     | PLL0_SR[LOLF]   PLL1_SR[LOLF] | PLLDIG  | PLL Loss of Lock Flags                                                          |
| 0x02D0 | 45     | SIU_OSR[OVF15:OVF0]           | SIU     | SIU combined overrun interrupt requests of the external interrupt Overrun Flags |
| 0x02E0 | 46     | SIU_EIISR[EIF0]               | SIU     | SIU External Interrupt Flag 0                                                   |
| 0x02F0 | 47     | SIU_EIISR[EIF1]               | SIU     | SIU External Interrupt Flag 1                                                   |
| 0x0300 | 48     | SIU_EIISR[EIF2]               | SIU     | SIU External Interrupt Flag 2                                                   |
| 0x0310 | 49     | SIU_EIISR[EIF3]               | SIU     | SIU External Interrupt Flag 3                                                   |
| 0x0320 | 50     | SIU_EIISR[EIF15:EIF4]         | SIU     | SIU External Interrupt Flags 15-4                                               |
| 0x0330 | 51     | eMIOS_0_GFLAG[F0]             | eMIOS_0 | eMIOS_0 channel 0 Flag                                                          |
| 0x0340 | 52     | eMIOS_0_GFLAG[F1]             | eMIOS_0 | eMIOS_0 channel 1 Flag                                                          |
| 0x0350 | 53     | eMIOS_0_GFLAG[F2]             | eMIOS_0 | eMIOS_0 channel 2 Flag                                                          |
| 0x0360 | 54     | eMIOS_0_GFLAG[F3]             | eMIOS_0 | eMIOS_0 channel 3 Flag                                                          |
| 0x0370 | 55     | eMIOS_0_GFLAG[F4]             | eMIOS_0 | eMIOS_0 channel 4 Flag                                                          |

Table continues on the next page...

**Table 4-4. Interrupt vectors and request sources (continued)**

| Offset | Vector | Source                                                                                                                 | Module   | Description                               |
|--------|--------|------------------------------------------------------------------------------------------------------------------------|----------|-------------------------------------------|
| 0x0380 | 56     | eMIOS_0_GFLAG[F5]                                                                                                      | eMIOS_0  | eMIOS_0 channel 5 Flag                    |
| 0x0390 | 57     | eMIOS_0_GFLAG[F6]                                                                                                      | eMIOS_0  | eMIOS_0 channel 6 Flag                    |
| 0x03A0 | 58     | eMIOS_0_GFLAG[F7]                                                                                                      | eMIOS_0  | eMIOS_0 channel 7 Flag                    |
| 0x03B0 | 59     | eMIOS_1_GFLAG[F0]                                                                                                      | eMIOS_1  | eMIOS_1 channel 0 Flag                    |
| 0x03C0 | 60     | eMIOS_1_GFLAG[F1]                                                                                                      | eMIOS_1  | eMIOS_1 channel 1 Flag                    |
| 0x03D0 | 61     | eMIOS_1_GFLAG[F2]                                                                                                      | eMIOS_1  | eMIOS_1 channel 2 Flag                    |
| 0x03E0 | 62     | eMIOS_1_GFLAG[F3]                                                                                                      | eMIOS_1  | eMIOS_1 channel 3 Flag                    |
| 0x03F0 | 63     | eMIOS_1_GFLAG[F4]                                                                                                      | eMIOS_1  | eMIOS_1 channel 4 Flag                    |
| 0x0400 | 64     | eMIOS_1_GFLAG[F5]                                                                                                      | eMIOS_1  | eMIOS_1 channel 5 Flag                    |
| 0x0410 | 65     | eMIOS_1_GFLAG[F6]                                                                                                      | eMIOS_1  | eMIOS_1 channel 6 Flag                    |
| 0x0420 | 66     | eMIOS_1_GFLAG[F7]                                                                                                      | eMIOS_1  | eMIOS_1 channel 7 Flag                    |
| 0x0430 | 67     | eTPU_SCR_MCR[SDMERR,SSAE<br>RR,SCMERR,WDT01/2,MGE1/2,IL<br>F1/2,LTO1/2,SCMMISF] and<br>eTPU_MESR_MESR[DCERR,CCE<br>RR] | eTPU_A/B | eTPU Engine A and B Global Exception      |
| 0x0440 | 68     | eTPU_GCR_CISR_ENG1[CIS0]                                                                                               | eTPU_A   | eTPU Engine A Channel 0 Interrupt Status  |
| 0x0450 | 69     | eTPU_GCR_CISR_ENG1[CIS1]                                                                                               | eTPU_A   | eTPU Engine A Channel 1 Interrupt Status  |
| 0x0460 | 70     | eTPU_GCR_CISR_ENG1[CIS2]                                                                                               | eTPU_A   | eTPU Engine A Channel 2 Interrupt Status  |
| 0x0470 | 71     | eTPU_GCR_CISR_ENG1[CIS3]                                                                                               | eTPU_A   | eTPU Engine A Channel 3 Interrupt Status  |
| 0x0480 | 72     | eTPU_GCR_CISR_ENG1[CIS4]                                                                                               | eTPU_A   | eTPU Engine A Channel 4 Interrupt Status  |
| 0x0490 | 73     | eTPU_GCR_CISR_ENG1[CIS5]                                                                                               | eTPU_A   | eTPU Engine A Channel 5 Interrupt Status  |
| 0x04A0 | 74     | eTPU_GCR_CISR_ENG1[CIS6]                                                                                               | eTPU_A   | eTPU Engine A Channel 6 Interrupt Status  |
| 0x04B0 | 75     | eTPU_GCR_CISR_ENG1[CIS7]                                                                                               | eTPU_A   | eTPU Engine A Channel 7 Interrupt Status  |
| 0x04C0 | 76     | eTPU_GCR_CISR_ENG1[CIS8]                                                                                               | eTPU_A   | eTPU Engine A Channel 8 Interrupt Status  |
| 0x04D0 | 77     | eTPU_GCR_CISR_ENG1[CIS9]                                                                                               | eTPU_A   | eTPU Engine A Channel 9 Interrupt Status  |
| 0x04E0 | 78     | eTPU_GCR_CISR_ENG1[CIS10]                                                                                              | eTPU_A   | eTPU Engine A Channel 10 Interrupt Status |
| 0x04F0 | 79     | eTPU_GCR_CISR_ENG1[CIS11]                                                                                              | eTPU_A   | eTPU Engine A Channel 11 Interrupt Status |
| 0x0500 | 80     | eTPU_GCR_CISR_ENG1[CIS12]                                                                                              | eTPU_A   | eTPU Engine A Channel 12 Interrupt Status |
| 0x0510 | 81     | eTPU_GCR_CISR_ENG1[CIS13]                                                                                              | eTPU_A   | eTPU Engine A Channel 13 Interrupt Status |
| 0x0520 | 82     | eTPU_GCR_CISR_ENG1[CIS14]                                                                                              | eTPU_A   | eTPU Engine A Channel 14 Interrupt Status |
| 0x0530 | 83     | eTPU_GCR_CISR_ENG1[CIS15]                                                                                              | eTPU_A   | eTPU Engine A Channel 15 Interrupt Status |
| 0x0540 | 84     | eTPU_GCR_CISR_ENG1[CIS16]                                                                                              | eTPU_A   | eTPU Engine A Channel 16 Interrupt Status |
| 0x0550 | 85     | eTPU_GCR_CISR_ENG1[CIS17]                                                                                              | eTPU_A   | eTPU Engine A Channel 17 Interrupt Status |
| 0x0560 | 86     | eTPU_GCR_CISR_ENG1[CIS18]                                                                                              | eTPU_A   | eTPU Engine A Channel 18 Interrupt Status |
| 0x0570 | 87     | eTPU_GCR_CISR_ENG1[CIS19]                                                                                              | eTPU_A   | eTPU Engine A Channel 19 Interrupt Status |
| 0x0580 | 88     | eTPU_GCR_CISR_ENG1[CIS20]                                                                                              | eTPU_A   | eTPU Engine A Channel 20 Interrupt Status |
| 0x0590 | 89     | eTPU_GCR_CISR_ENG1[CIS21]                                                                                              | eTPU_A   | eTPU Engine A Channel 21 Interrupt Status |
| 0x05A0 | 90     | eTPU_GCR_CISR_ENG1[CIS22]                                                                                              | eTPU_A   | eTPU Engine A Channel 22 Interrupt Status |
| 0x05B0 | 91     | eTPU_GCR_CISR_ENG1[CIS23]                                                                                              | eTPU_A   | eTPU Engine A Channel 23 Interrupt Status |
| 0x05C0 | 92     | eTPU_GCR_CISR_ENG1[CIS24]                                                                                              | eTPU_A   | eTPU Engine A Channel 24 Interrupt Status |

Table continues on the next page...

**Table 4-4. Interrupt vectors and request sources (continued)**

| Offset | Vector | Source                                                                                                | Module  | Description                                                                                                                                |
|--------|--------|-------------------------------------------------------------------------------------------------------|---------|--------------------------------------------------------------------------------------------------------------------------------------------|
| 0x05D0 | 93     | eTPU_GCR_CISR_ENG1[CIS25]                                                                             | eTPU_A  | eTPU Engine A Channel 25 Interrupt Status                                                                                                  |
| 0x05E0 | 94     | eTPU_GCR_CISR_ENG1[CIS26]                                                                             | eTPU_A  | eTPU Engine A Channel 26 Interrupt Status                                                                                                  |
| 0x05F0 | 95     | eTPU_GCR_CISR_ENG1[CIS27]                                                                             | eTPU_A  | eTPU Engine A Channel 27 Interrupt Status                                                                                                  |
| 0x0600 | 96     | eTPU_GCR_CISR_ENG1[CIS28]                                                                             | eTPU_A  | eTPU Engine A Channel 28 Interrupt Status                                                                                                  |
| 0x0610 | 97     | eTPU_GCR_CISR_ENG1[CIS29]                                                                             | eTPU_A  | eTPU Engine A Channel 29 Interrupt Status                                                                                                  |
| 0x0620 | 98     | eTPU_GCR_CISR_ENG1[CIS30]                                                                             | eTPU_A  | eTPU Engine A Channel 30 Interrupt Status                                                                                                  |
| 0x0630 | 99     | eTPU_GCR_CISR_ENG1[CIS31]                                                                             | eTPU_A  | eTPU Engine A Channel 31 Interrupt Status                                                                                                  |
| 0x0640 | 100    | EQADCA_FISR <sub>x</sub> [TORF]<br>EQADCA_FISR <sub>x</sub> [RFOF]<br>EQADCA_FISR <sub>x</sub> [CFUF] | eQADC_A | eQADC combined overrun interrupt requests from all of the FIFOs:<br><br>Trigger Overrun, Receive FIFO Overflow, and command FIFO Underflow |
| 0x0650 | 101    | EQADCA_FISR0[NCF]                                                                                     | eQADC_A | eQADC command FIFO 0 Non-Coherency Flag                                                                                                    |
| 0x0660 | 102    | EQADCA_FISR0[PF]                                                                                      | eQADC_A | eQADC command FIFO 0 Pause Flag                                                                                                            |
| 0x0670 | 103    | EQADCA_FISR0[EOQF]                                                                                    | eQADC_A | eQADC command FIFO 0 command queue End of Queue Flag                                                                                       |
| 0x0680 | 104    | EQADCA_FISR0[CFFF]                                                                                    | eQADC_A | eQADC Command FIFO 0 Fill Flag                                                                                                             |
| 0x0690 | 105    | EQADCA_FISR0[RFDF]                                                                                    | eQADC_A | eQADC Receive FIFO 0 Drain Flag                                                                                                            |
| 0x06A0 | 106    | EQADCA_FISR1[NCF]                                                                                     | eQADC_A | eQADC command FIFO 1 Non-Coherency Flag                                                                                                    |
| 0x06B0 | 107    | EQADCA_FISR1[PF]                                                                                      | eQADC_A | eQADC command FIFO 1 Pause Flag                                                                                                            |
| 0x06C0 | 108    | EQADCA_FISR1[EOQF]                                                                                    | eQADC_A | eQADC command FIFO 1 command queue End of Queue Flag                                                                                       |
| 0x06D0 | 109    | EQADCA_FISR1[CFFF]                                                                                    | eQADC_A | eQADC Command FIFO 1 Fill Flag                                                                                                             |
| 0x06E0 | 110    | EQADCA_FISR1[RFDF]                                                                                    | eQADC_A | eQADC Receive FIFO 1 Drain Flag                                                                                                            |
| 0x06F0 | 111    | EQADCA_FISR2[NCF]                                                                                     | eQADC_A | eQADC command FIFO 2 Non-Coherency Flag                                                                                                    |
| 0x0700 | 112    | EQADCA_FISR2[PF]                                                                                      | eQADC_A | eQADC command FIFO 2 Pause Flag                                                                                                            |
| 0x0710 | 113    | EQADCA_FISR2[EOQF]                                                                                    | eQADC_A | eQADC command FIFO 2 command queue End of Queue Flag                                                                                       |
| 0x0720 | 114    | EQADCA_FISR2[CFFF]                                                                                    | eQADC_A | eQADC Command FIFO 2 Fill Flag                                                                                                             |
| 0x0730 | 115    | EQADCA_FISR2[RFDF]                                                                                    | eQADC_A | eQADC Receive FIFO 2 Drain Flag                                                                                                            |
| 0x0740 | 116    | EQADCA_FISR3[NCF]                                                                                     | eQADC_A | eQADC command FIFO 3 Non-Coherency Flag                                                                                                    |
| 0x0750 | 117    | EQADCA_FISR3[PF]                                                                                      | eQADC_A | eQADC command FIFO 3 Pause Flag                                                                                                            |
| 0x0760 | 118    | EQADCA_FISR3[EOQF]                                                                                    | eQADC_A | eQADC command FIFO 3 command queue End of Queue Flag                                                                                       |
| 0x0770 | 119    | EQADCA_FISR3[CFFF]                                                                                    | eQADC_A | eQADC Command FIFO 3 Fill Flag                                                                                                             |
| 0x0780 | 120    | EQADCA_FISR3[RFDF]                                                                                    | eQADC_A | eQADC Receive FIFO 3 Drain Flag                                                                                                            |
| 0x0790 | 121    | EQADCA_FISR4[NCF]                                                                                     | eQADC_A | eQADC command FIFO 4 Non-Coherency Flag                                                                                                    |
| 0x07A0 | 122    | EQADCA_FISR4[PF]                                                                                      | eQADC_A | eQADC command FIFO 4 Pause Flag                                                                                                            |

Table continues on the next page...

**Table 4-4. Interrupt vectors and request sources (continued)**

| Offset | Vector | Source                                                               | Module  | Description                                                                                                                                     |
|--------|--------|----------------------------------------------------------------------|---------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x07B0 | 123    | EQADCA_FISR4[EOQF]                                                   | eQADC_A | eQADC command FIFO 4 command queue End of Queue Flag                                                                                            |
| 0x07C0 | 124    | EQADCA_FISR4[CFFF]                                                   | eQADC_A | eQADC Command FIFO 4 Fill Flag                                                                                                                  |
| 0x07D0 | 125    | EQADCA_FISR4[RFDF]                                                   | eQADC_A | eQADC Receive FIFO 4 Drain Flag                                                                                                                 |
| 0x07E0 | 126    | EQADCA_FISR5[NCF]                                                    | eQADC_A | eQADC command FIFO 5 Non-Coherency Flag                                                                                                         |
| 0x07F0 | 127    | EQADCA_FISR5[PF]                                                     | eQADC_A | eQADC command FIFO 5 Pause Flag                                                                                                                 |
| 0x0800 | 128    | EQADCA_FISR5[EOQF]                                                   | eQADC_A | eQADC command FIFO 5 command queue End of Queue Flag                                                                                            |
| 0x0810 | 129    | EQADCA_FISR5[CFFF]                                                   | eQADC_A | eQADC Command FIFO 5 Fill Flag                                                                                                                  |
| 0x0820 | 130    | EQADCA_FISR5[RFDF]                                                   | eQADC_A | eQADC Receive FIFO 5 Drain Flag                                                                                                                 |
| 0x0830 | 131    | DSPIB_SR[TFUF]<br>DSPIB_SR[RFOF]<br>DSPIB_SR[SPEF]<br>DSPIB_SR[DPEF] | DSPI_B  | DSPI_B combined overrun and parity error interrupt requests:<br>Transmit FIFO Underflow/Receive FIFO Overflow SPI Parity Error/DSI Parity Error |
| 0x0840 | 132    | DSPIB_SR[EOQF]                                                       | DSPI_B  | DSPI_B transmit FIFO End of Queue Flag                                                                                                          |
| 0x0850 | 133    | DSPIB_SR[TFFF]                                                       | DSPI_B  | DSPI_B Transmit FIFO Fill Flag                                                                                                                  |
| 0x0860 | 134    | DSPIB_SR[TCF]   DSPIB_SR[DDIF]                                       | DSPI_B  | DSPI_B Transfer Complete/DSI Data Match Flag                                                                                                    |
| 0x0870 | 135    | DSPIB_SR[RFDF]                                                       | DSPI_B  | DSPI_B Receive FIFO Drain Flag                                                                                                                  |
| 0x0880 | 136    | DSPIC_SR[TFUF]<br>DSPIC_SR[RFOF]<br>DSPIC_SR[SPEF]<br>DSPIC_SR[DPEF] | DSPI_C  | DSPI_C combined overrun and parity error interrupt requests:<br>Transmit FIFO Underflow/Receive FIFO Overflow SPI Parity Error/DSI Parity Error |
| 0x0890 | 137    | DSPIC_SR[EOQF]                                                       | DSPI_C  | DSPI_C transmit FIFO End of Queue Flag                                                                                                          |
| 0x08A0 | 138    | DSPIC_SR[TFFF]                                                       | DSPI_C  | DSPI_C Transmit FIFO Fill Flag                                                                                                                  |
| 0x08B0 | 139    | DSPIC_SR[TCF]   DSPIC_SR[DDIF]                                       | DSPI_C  | DSPI_C Transfer Complete/DSI Data Match Flag                                                                                                    |
| 0x08C0 | 140    | DSPIC_SR[RFDF]                                                       | DSPI_C  | DSPI_C Receive FIFO Drain Flag                                                                                                                  |
| 0x08D0 | 141    | DSPID_SR[TFUF]<br>DSPID_SR[RFOF]<br>DSPID_SR[SPEF]<br>DSPID_SR[DPEF] | DSPI_D  | DSPI_D combined overrun and parity error interrupt requests:<br>Transmit FIFO Underflow/Receive FIFO Overflow SPI Parity Error/DSI Parity Error |
| 0x08E0 | 142    | DSPID_SR[EOQF]                                                       | DSPI_D  | DSPI_D transmit FIFO End of Queue Flag                                                                                                          |
| 0x08F0 | 143    | DSPID_SR[TFFF]                                                       | DSPI_D  | DSPI_D Transmit FIFO Fill Flag                                                                                                                  |
| 0x0900 | 144    | DSPID_SR[TCF]   DSPID_SR[DDIF]                                       | DSPI_D  | DSPI_D Transfer Complete/DSI Data Match Flag                                                                                                    |
| 0x0910 | 145    | DSPID_SR[RFDF]                                                       | DSPI_D  | DSPI_D Receive FIFO Drain Flag                                                                                                                  |
| 0x0920 | 146    | ESCIA_SR[TDRE]<br>ESCIA_SR[TC]                                       | eSCI_A  | Combined Interrupt Requests of ESCI Module A:                                                                                                   |

Table continues on the next page...

**Table 4-4. Interrupt vectors and request sources (continued)**

| Offset | Vector | Source                                                                                                                                                                                                                                                                                                                    | Module | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|--------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|        |        | ESCIA_SR[RDRF]<br>ESCIA_SR[IDLE]<br>ESCIA_SR[OR]<br>ESCIA_SR[NF]<br>ESCIA_SR[FE]<br>ESCIA_SR[PF]<br>ESCIA_SR[BERR]<br>ESCIA_SR[RXRDY]<br>ESCIA_SR[TXRDY]<br>ESCIA_SR[LWAKE]<br>ESCIA_SR[STO]<br>ESCIA_SR[PBERR]<br>ESCIA_SR[CERR]<br>ESCIA_SR[CKERR]<br>ESCIA_SR[FRC]<br>ESCIA_SR[OVFL]                                   |        | Transmit Data Register Empty, Transmit Complete, Receive Data Register Full, Idle line, Overrun, Noise Flag, Framing Error Flag, and Parity Error Flag interrupt requests, SCI Status Register 2 Bit Error interrupt request, LIN Status Register 1 Receive Data Ready, Transmit Data Ready, Received LIN Wakeup Signal, Slave TimeOut, Physical Bus Error, CRC Error, Checksum Error, Frame Complete interrupts requests, and LIN Status Register 2 Receive Register Overflow                                                      |
| 0x0930 | 147    | Master0 Snoop ipi_int                                                                                                                                                                                                                                                                                                     | PCU    | PCU_IR0[OIF]   PCU_IR0(EIF)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 0x0940 | 148    | Master1 Snoop ipi_int                                                                                                                                                                                                                                                                                                     | PCU    | PCU_IR1[OIF]   PCU_IR1(EIF)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 0x0950 | 149    | ESCIB_SR[TDRE]<br>ESCIB_SR[TC]<br>ESCIB_SR[RDRF]<br>ESCIB_SR[IDLE]<br>ESCIB_SR[OR]<br>ESCIB_SR[NF]<br>ESCIB_SR[FE]<br>ESCIB_SR[PF]<br>ESCIB_SR[BERR]<br>ESCIB_SR[RXRDY]<br>ESCIB_SR[TXRDY]<br>ESCIB_SR[LWAKE]<br>ESCIB_SR[STO]<br>ESCIB_SR[PBERR]<br>ESCIB_SR[CERR]<br>ESCIB_SR[CKERR]<br>ESCIB_SR[FRC]<br>ESCIB_SR[OVFL] | eSCI_B | Combined Interrupt Requests of ESCI Module B:<br><br>Transmit Data Register Empty, Transmit Complete, Receive Data Register Full, Idle line, Overrun, Noise Flag, Framing Error Flag, and Parity Error Flag interrupt requests, SCI Status Register 2 Bit Error interrupt request, LIN Status Register 1 Receive Data Ready, Transmit Data Ready, Received LIN Wakeup Signal, Slave TimeOut, Physical Bus Error, CRC Error, Checksum Error, Frame Complete interrupts requests, and LIN Status Register 2 Receive Register Overflow |
| 0x0960 | 150    | Reserved                                                                                                                                                                                                                                                                                                                  | —      | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 0x0970 | 151    | Reserved                                                                                                                                                                                                                                                                                                                  | —      | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

Table continues on the next page...

**Table 4-4. Interrupt vectors and request sources (continued)**

| Offset | Vector | Source                                                                                              | Module    | Description                                                                                                                                       |
|--------|--------|-----------------------------------------------------------------------------------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x0980 | 152    | CANA_ESR1[BOFF_INT]   [TWRNINT]   [RWRNINT]                                                         | FlexCAN_A | FlexCAN_A Bus Off, Transmit Warning, Receive Warning                                                                                              |
| 0x0990 | 153    | CANA_ESR1[ERR_INT]   CANA_ERRSR[ECC_CE_INT]   CANA_ERRSR[ECC_HANCE_INT]   CANA_ERRSR[ECC_FANCE_INT] | FlexCAN_A | FlexCAN_A Error, FlexCAN_A ECC Correctable Error, FlexCAN_A ECC Host Access Non-Correctable Error, FlexCAN_A ECC CAN Access Non-Correctable Error |
| 0x09A0 | 154    | Reserved                                                                                            | FlexCAN_A | Reserved                                                                                                                                          |
| 0x09B0 | 155    | CANA_IFRL[BUF0]                                                                                     | FlexCAN_A | FlexCAN_A Buffer 0 Interrupt                                                                                                                      |
| 0x09C0 | 156    | CANA_IFRL[BUF1]                                                                                     | FlexCAN_A | FlexCAN_A Buffer 1 Interrupt                                                                                                                      |
| 0x09D0 | 157    | CANA_IFRL[BUF2]                                                                                     | FlexCAN_A | FlexCAN_A Buffer 2 Interrupt                                                                                                                      |
| 0x09E0 | 158    | CANA_IFRL[BUF3]                                                                                     | FlexCAN_A | FlexCAN_A Buffer 3 Interrupt                                                                                                                      |
| 0x09F0 | 159    | CANA_IFRL[BUF4]                                                                                     | FlexCAN_A | FlexCAN_A Buffer 4 Interrupt                                                                                                                      |
| 0x0A00 | 160    | CANA_IFRL[BUF5]                                                                                     | FlexCAN_A | FlexCAN_A Buffer 5 Interrupt                                                                                                                      |
| 0x0A10 | 161    | CANA_IFRL[BUF6]                                                                                     | FlexCAN_A | FlexCAN_A Buffer 6 Interrupt                                                                                                                      |
| 0x0A20 | 162    | CANA_IFRL[BUF7]                                                                                     | FlexCAN_A | FlexCAN_A Buffer 7 Interrupt                                                                                                                      |
| 0x0A30 | 163    | CANA_IFRL[BUF8]                                                                                     | FlexCAN_A | FlexCAN_A Buffer 8 Interrupt                                                                                                                      |
| 0x0A40 | 164    | CANA_IFRL[BUF9]                                                                                     | FlexCAN_A | FlexCAN_A Buffer 9 Interrupt                                                                                                                      |
| 0x0A50 | 165    | CANA_IFRL[BUF10]                                                                                    | FlexCAN_A | FlexCAN_A Buffer 10 Interrupt                                                                                                                     |
| 0x0A60 | 166    | CANA_IFRL[BUF11]                                                                                    | FlexCAN_A | FlexCAN_A Buffer 11 Interrupt                                                                                                                     |
| 0x0A70 | 167    | CANA_IFRL[BUF12]                                                                                    | FlexCAN_A | FlexCAN_A Buffer 12 Interrupt                                                                                                                     |
| 0x0A80 | 168    | CANA_IFRL[BUF13]                                                                                    | FlexCAN_A | FlexCAN_A Buffer 13 Interrupt                                                                                                                     |
| 0x0A90 | 169    | CANA_IFRL[BUF14]                                                                                    | FlexCAN_A | FlexCAN_A Buffer 14 Interrupt                                                                                                                     |
| 0x0AA0 | 170    | CANA_IFRL[BUF15]                                                                                    | FlexCAN_A | FlexCAN_A Buffer 15 Interrupt                                                                                                                     |
| 0x0AB0 | 171    | CANA_IFRL[BUF31:BUF16]                                                                              | FlexCAN_A | FlexCAN_A Buffers 31-16 Interrupts                                                                                                                |
| 0x0AC0 | 172    | CANA_IFRH[BUF63:BUF32]                                                                              | FlexCAN_A | FlexCAN_A Buffers 63-32 Interrupts                                                                                                                |
| 0x0AD0 | 173    | CANC_ESR1[BOFF_INT]   [TWRNINT]   [RWRNINT]                                                         | FlexCAN_C | FlexCAN_C Bus Off, Transmit Warning, Receive Warning                                                                                              |
| 0x0AE0 | 174    | CANC_ESR1[ERR_INT]   CANC_ERRSR[ECC_CE_INT]   CANC_ERRSR[ECC_HANCE_INT]   CANC_ERRSR[ECC_FANCE_INT] | FlexCAN_C | FlexCAN_C Error, FlexCAN_C ECC Correctable Error, FlexCAN_C ECC Host Access Non-Correctable Error, FlexCAN_C ECC CAN Access Non-Correctable Error |
| 0x0AF0 | 175    | Reserved                                                                                            | FlexCAN_C | Reserved                                                                                                                                          |
| 0x0B00 | 176    | CANC_IFRL[BUF0]                                                                                     | FlexCAN_C | FlexCAN_C Buffer 0 Interrupt                                                                                                                      |
| 0x0B10 | 177    | CANC_IFRL[BUF1]                                                                                     | FlexCAN_C | FlexCAN_C Buffer 1 Interrupt                                                                                                                      |
| 0x0B20 | 178    | CANC_IFRL[BUF2]                                                                                     | FlexCAN_C | FlexCAN_C Buffer 2 Interrupt                                                                                                                      |
| 0x0B30 | 179    | CANC_IFRL[BUF3]                                                                                     | FlexCAN_C | FlexCAN_C Buffer 3 Interrupt                                                                                                                      |
| 0x0B40 | 180    | CANC_IFRL[BUF4]                                                                                     | FlexCAN_C | FlexCAN_C Buffer 4 Interrupt                                                                                                                      |
| 0x0B50 | 181    | CANC_IFRL[BUF5]                                                                                     | FlexCAN_C | FlexCAN_C Buffer 5 Interrupt                                                                                                                      |
| 0x0B60 | 182    | CANC_IFRL[BUF6]                                                                                     | FlexCAN_C | FlexCAN_C Buffer 6 Interrupt                                                                                                                      |
| 0x0B70 | 183    | CANC_IFRL[BUF7]                                                                                     | FlexCAN_C | FlexCAN_C Buffer 7 Interrupt                                                                                                                      |
| 0x0B80 | 184    | CANC_IFRL[BUF8]                                                                                     | FlexCAN_C | FlexCAN_C Buffer 8 Interrupt                                                                                                                      |
| 0x0B90 | 185    | CANC_IFRL[BUF9]                                                                                     | FlexCAN_C | FlexCAN_C Buffer 9 Interrupt                                                                                                                      |

Table continues on the next page...

**Table 4-4. Interrupt vectors and request sources (continued)**

| Offset | Vector | Source                                                                                                                             | Module    | Description                                                                                                                                                                                                                                                                                                                    |
|--------|--------|------------------------------------------------------------------------------------------------------------------------------------|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x0BA0 | 186    | CANC_IFRL[BUF10]                                                                                                                   | FlexCAN_C | FlexCAN_C Buffer 10 Interrupt                                                                                                                                                                                                                                                                                                  |
| 0x0BB0 | 187    | CANC_IFRL[BUF11]                                                                                                                   | FlexCAN_C | FlexCAN_C Buffer 11 Interrupt                                                                                                                                                                                                                                                                                                  |
| 0x0BC0 | 188    | CANC_IFRL[BUF12]                                                                                                                   | FlexCAN_C | FlexCAN_C Buffer 12 Interrupt                                                                                                                                                                                                                                                                                                  |
| 0x0BD0 | 189    | CANC_IFRL[BUF13]                                                                                                                   | FlexCAN_C | FlexCAN_C Buffer 13 Interrupt                                                                                                                                                                                                                                                                                                  |
| 0x0BE0 | 190    | CANC_IFRL[BUF14]                                                                                                                   | FlexCAN_C | FlexCAN_C Buffer 14 Interrupt                                                                                                                                                                                                                                                                                                  |
| 0x0BF0 | 191    | CANC_IFRL[BUF15]                                                                                                                   | FlexCAN_C | FlexCAN_C Buffer 15 Interrupt                                                                                                                                                                                                                                                                                                  |
| 0x0C00 | 192    | CANC_IFRL[BUF31:BUF16]                                                                                                             | FlexCAN_C | FlexCAN_C Buffers 31-16 Interrupts                                                                                                                                                                                                                                                                                             |
| 0x0C10 | 193    | CANC_IFRH[BUF63:BUF32]                                                                                                             | FlexCAN_C | FlexCAN_C Buffers 63-32 Interrupts                                                                                                                                                                                                                                                                                             |
| 0x0C20 | 194    | EIR[TXF]                                                                                                                           | FEC       | FEC Transmit Frame flag                                                                                                                                                                                                                                                                                                        |
| 0x0C30 | 195    | EIR[RXF]                                                                                                                           | FEC       | FEC Receive Frame flag                                                                                                                                                                                                                                                                                                         |
| 0x0C40 | 196    | EIR[HBERR]   EIR[BABR]  <br>EIR[BABT]   EIR[GRA]   EIR[TXB]  <br>EIR[RXB]   EIR[MII]   EIR[EBERR]  <br>EIR[LC]   EIR[RL]   EIR[UN] | FEC       | Combined Interrupt Requests of the FEC Ethernet Interrupt Event Register:<br><br>Heartbeat Error, Babbling Receive Error, Babbling Transmit Error, Graceful Stop Complete, Transmit Buffer, Receive Buffer, Media Independent Interface, Ethernet Bus Error, Late Collision, Collision Retry Limit, and Transmit FIFO Underrun |
| 0x0C50 | 197    | Reserved                                                                                                                           | —         | Reserved                                                                                                                                                                                                                                                                                                                       |
| 0x0C60 | 198    | Reserved                                                                                                                           | —         | Reserved                                                                                                                                                                                                                                                                                                                       |
| 0x0C70 | 199    | Reserved                                                                                                                           | —         | Reserved                                                                                                                                                                                                                                                                                                                       |
| 0x0C80 | 200    | STM_CIR0[CIF]                                                                                                                      | STM       | System Timer Module Interrupt 0                                                                                                                                                                                                                                                                                                |
| 0x0C90 | 201    | STM_CIR1[CIF], STM_CIR2[CIF],<br>STM_CIR3[CIF]                                                                                     | STM       | System Timer Module Interrupts 1, 2, 3                                                                                                                                                                                                                                                                                         |
| 0x0CA0 | 202    | eMIOS_0_GFLAG[F16]                                                                                                                 | eMIOS_0   | eMIOS_0 channel 16 Flag                                                                                                                                                                                                                                                                                                        |
| 0x0CB0 | 203    | eMIOS_0_GFLAG[F17]                                                                                                                 | eMIOS_0   | eMIOS_0 channel 17 Flag                                                                                                                                                                                                                                                                                                        |
| 0x0CC0 | 204    | eMIOS_0_GFLAG[F18]                                                                                                                 | eMIOS_0   | eMIOS_0 channel 18 Flag                                                                                                                                                                                                                                                                                                        |
| 0x0CD0 | 205    | eMIOS_0_GFLAG[F19]                                                                                                                 | eMIOS_0   | eMIOS_0 channel 19 Flag                                                                                                                                                                                                                                                                                                        |
| 0x0CE0 | 206    | eMIOS_0_GFLAG[F20]                                                                                                                 | eMIOS_0   | eMIOS_0 channel 20 Flag                                                                                                                                                                                                                                                                                                        |
| 0x0CF0 | 207    | eMIOS_0_GFLAG[F21]                                                                                                                 | eMIOS_0   | eMIOS_0 channel 21 Flag                                                                                                                                                                                                                                                                                                        |
| 0x0D00 | 208    | eMIOS_0_GFLAG[F22]                                                                                                                 | eMIOS_0   | eMIOS_0 channel 22 Flag                                                                                                                                                                                                                                                                                                        |
| 0x0D10 | 209    | eMIOS_0_GFLAG[F23]                                                                                                                 | eMIOS_0   | eMIOS_0 channel 23 Flag                                                                                                                                                                                                                                                                                                        |
| 0x0D20 | 210    | EDMAA_ERRH[ERR63:ERR32]                                                                                                            | eDMA_A    | eDMA channel Error flags 32-63                                                                                                                                                                                                                                                                                                 |
| 0x0D30 | 211    | EDMAA_IRQRH[INT32]                                                                                                                 | eDMA_A    | eDMA channel Interrupt 32                                                                                                                                                                                                                                                                                                      |
| 0x0D40 | 212    | EDMAA_IRQRH[INT33]                                                                                                                 | eDMA_A    | eDMA channel Interrupt 33                                                                                                                                                                                                                                                                                                      |
| 0x0D50 | 213    | EDMAA_IRQRH[INT34]                                                                                                                 | eDMA_A    | eDMA channel Interrupt 34                                                                                                                                                                                                                                                                                                      |
| 0x0D60 | 214    | EDMAA_IRQRH[INT35]                                                                                                                 | eDMA_A    | eDMA channel Interrupt 35                                                                                                                                                                                                                                                                                                      |
| 0x0D70 | 215    | EDMAA_IRQRH[INT36]                                                                                                                 | eDMA_A    | eDMA channel Interrupt 36                                                                                                                                                                                                                                                                                                      |
| 0x0D80 | 216    | EDMAA_IRQRH[INT37]                                                                                                                 | eDMA_A    | eDMA channel Interrupt 37                                                                                                                                                                                                                                                                                                      |
| 0x0D90 | 217    | EDMAA_IRQRH[INT38]                                                                                                                 | eDMA_A    | eDMA channel Interrupt 38                                                                                                                                                                                                                                                                                                      |
| 0x0DA0 | 218    | EDMAA_IRQRH[INT39]                                                                                                                 | eDMA_A    | eDMA channel Interrupt 39                                                                                                                                                                                                                                                                                                      |

Table continues on the next page...

**Table 4-4. Interrupt vectors and request sources (continued)**

| Offset | Vector | Source                    | Module | Description                               |
|--------|--------|---------------------------|--------|-------------------------------------------|
| 0x0DB0 | 219    | EDMAA_IRQRH[INT40]        | eDMA_A | eDMA channel Interrupt 40                 |
| 0x0DC0 | 220    | EDMAA_IRQRH[INT41]        | eDMA_A | eDMA channel Interrupt 41                 |
| 0x0DD0 | 221    | EDMAA_IRQRH[INT42]        | eDMA_A | eDMA channel Interrupt 42                 |
| 0x0DE0 | 222    | EDMAA_IRQRH[INT43]        | eDMA_A | eDMA channel Interrupt 43                 |
| 0x0DF0 | 223    | EDMAA_IRQRH[INT44]        | eDMA_A | eDMA channel Interrupt 44                 |
| 0x0E00 | 224    | EDMAA_IRQRH[INT45]        | eDMA_A | eDMA channel Interrupt 45                 |
| 0x0E10 | 225    | EDMAA_IRQRH[INT46]        | eDMA_A | eDMA channel Interrupt 46                 |
| 0x0E20 | 226    | EDMAA_IRQRH[INT47]        | eDMA_A | eDMA channel Interrupt 47                 |
| 0x0E30 | 227    | EDMAA_IRQRH[INT48]        | eDMA_A | eDMA channel Interrupt 48                 |
| 0x0E40 | 228    | EDMAA_IRQRH[INT49]        | eDMA_A | eDMA channel Interrupt 49                 |
| 0x0E50 | 229    | EDMAA_IRQRH[INT50]        | eDMA_A | eDMA channel Interrupt 50                 |
| 0x0E60 | 230    | EDMAA_IRQRH[INT51]        | eDMA_A | eDMA channel Interrupt 51                 |
| 0x0E70 | 231    | EDMAA_IRQRH[INT52]        | eDMA_A | eDMA channel Interrupt 52                 |
| 0x0E80 | 232    | EDMAA_IRQRH[INT53]        | eDMA_A | eDMA channel Interrupt 53                 |
| 0x0E90 | 233    | EDMAA_IRQRH[INT54]        | eDMA_A | eDMA channel Interrupt 54                 |
| 0x0EA0 | 234    | EDMAA_IRQRH[INT55]        | eDMA_A | eDMA channel Interrupt 55                 |
| 0x0EB0 | 235    | EDMAA_IRQRH[INT56]        | eDMA_A | eDMA channel Interrupt 56                 |
| 0x0EC0 | 236    | EDMAA_IRQRH[INT57]        | eDMA_A | eDMA channel Interrupt 57                 |
| 0x0ED0 | 237    | EDMAA_IRQRH[INT58]        | eDMA_A | eDMA channel Interrupt 58                 |
| 0x0EE0 | 238    | EDMAA_IRQRH[INT59]        | eDMA_A | eDMA channel Interrupt 59                 |
| 0x0EF0 | 239    | EDMAA_IRQRH[INT60]        | eDMA_A | eDMA channel Interrupt 60                 |
| 0x0F00 | 240    | EDMAA_IRQRH[INT61]        | eDMA_A | eDMA channel Interrupt 61                 |
| 0x0F10 | 241    | EDMAA_IRQRH[INT62]        | eDMA_A | eDMA channel Interrupt 62                 |
| 0x0F20 | 242    | EDMAA_IRQRH[INT63]        | eDMA_A | eDMA channel Interrupt 63                 |
| 0x0F30 | 243    | eTPU_GCR_CISR_ENG2[CIS0]  | eTPU_B | eTPU Engine B Channel 0 Interrupt Status  |
| 0x0F40 | 244    | eTPU_GCR_CISR_ENG2[CIS1]  | eTPU_B | eTPU Engine B Channel 1 Interrupt Status  |
| 0x0F50 | 245    | eTPU_GCR_CISR_ENG2[CIS2]  | eTPU_B | eTPU Engine B Channel 2 Interrupt Status  |
| 0x0F60 | 246    | eTPU_GCR_CISR_ENG2[CIS3]  | eTPU_B | eTPU Engine B Channel 3 Interrupt Status  |
| 0x0F70 | 247    | eTPU_GCR_CISR_ENG2[CIS4]  | eTPU_B | eTPU Engine B Channel 4 Interrupt Status  |
| 0x0F80 | 248    | eTPU_GCR_CISR_ENG2[CIS5]  | eTPU_B | eTPU Engine B Channel 5 Interrupt Status  |
| 0x0F90 | 249    | eTPU_GCR_CISR_ENG2[CIS6]  | eTPU_B | eTPU Engine B Channel 6 Interrupt Status  |
| 0x0FA0 | 250    | eTPU_GCR_CISR_ENG2[CIS7]  | eTPU_B | eTPU Engine B Channel 7 Interrupt Status  |
| 0x0FB0 | 251    | eTPU_GCR_CISR_ENG2[CIS8]  | eTPU_B | eTPU Engine B Channel 8 Interrupt Status  |
| 0x0FC0 | 252    | eTPU_GCR_CISR_ENG2[CIS9]  | eTPU_B | eTPU Engine B Channel 9 Interrupt Status  |
| 0x0fd0 | 253    | eTPU_GCR_CISR_ENG2[CIS10] | eTPU_B | eTPU Engine B Channel 10 Interrupt Status |
| 0x0fe0 | 254    | eTPU_GCR_CISR_ENG2[CIS11] | eTPU_B | eTPU Engine B Channel 11 Interrupt Status |
| 0x0ff0 | 255    | eTPU_GCR_CISR_ENG2[CIS12] | eTPU_B | eTPU Engine B Channel 12 Interrupt Status |
| 0x1000 | 256    | eTPU_GCR_CISR_ENG2[CIS13] | eTPU_B | eTPU Engine B Channel 13 Interrupt Status |
| 0x1010 | 257    | eTPU_GCR_CISR_ENG2[CIS14] | eTPU_B | eTPU Engine B Channel 14 Interrupt Status |

Table continues on the next page...



**Table 4-4. Interrupt vectors and request sources (continued)**

| Offset | Vector | Source                                                                                                       | Module    | Description                                                                                                                                         |
|--------|--------|--------------------------------------------------------------------------------------------------------------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x1020 | 258    | eTPU_GCR_CISR_ENG2[CIS15]                                                                                    | eTPU_B    | eTPU Engine B Channel 15 Interrupt Status                                                                                                           |
| 0x1030 | 259    | eTPU_GCR_CISR_ENG2[CIS16]                                                                                    | eTPU_B    | eTPU Engine B Channel 16 Interrupt Status                                                                                                           |
| 0x1040 | 260    | eTPU_GCR_CISR_ENG2[CIS17]                                                                                    | eTPU_B    | eTPU Engine B Channel 17 Interrupt Status                                                                                                           |
| 0x1050 | 261    | eTPU_GCR_CISR_ENG2[CIS18]                                                                                    | eTPU_B    | eTPU Engine B Channel 18 Interrupt Status                                                                                                           |
| 0x1060 | 262    | eTPU_GCR_CISR_ENG2[CIS19]                                                                                    | eTPU_B    | eTPU Engine B Channel 19 Interrupt Status                                                                                                           |
| 0x1070 | 263    | eTPU_GCR_CISR_ENG2[CIS20]                                                                                    | eTPU_B    | eTPU Engine B Channel 20 Interrupt Status                                                                                                           |
| 0x1080 | 264    | eTPU_GCR_CISR_ENG2[CIS21]                                                                                    | eTPU_B    | eTPU Engine B Channel 21 Interrupt Status                                                                                                           |
| 0x1090 | 265    | eTPU_GCR_CISR_ENG2[CIS22]                                                                                    | eTPU_B    | eTPU Engine B Channel 22 Interrupt Status                                                                                                           |
| 0x10A0 | 266    | eTPU_GCR_CISR_ENG2[CIS23]                                                                                    | eTPU_B    | eTPU Engine B Channel 23 Interrupt Status                                                                                                           |
| 0x10B0 | 267    | eTPU_GCR_CISR_ENG2[CIS24]                                                                                    | eTPU_B    | eTPU Engine B Channel 24 Interrupt Status                                                                                                           |
| 0x10C0 | 268    | eTPU_GCR_CISR_ENG2[CIS25]                                                                                    | eTPU_B    | eTPU Engine B Channel 25 Interrupt Status                                                                                                           |
| 0x10D0 | 269    | eTPU_GCR_CISR_ENG2[CIS26]                                                                                    | eTPU_B    | eTPU Engine B Channel 26 Interrupt Status                                                                                                           |
| 0x10E0 | 270    | eTPU_GCR_CISR_ENG2[CIS27]                                                                                    | eTPU_B    | eTPU Engine B Channel 27 Interrupt Status                                                                                                           |
| 0x10F0 | 271    | eTPU_GCR_CISR_ENG2[CIS28]                                                                                    | eTPU_B    | eTPU Engine B Channel 28 Interrupt Status                                                                                                           |
| 0x1100 | 272    | eTPU_GCR_CISR_ENG2[CIS29]                                                                                    | eTPU_B    | eTPU Engine B Channel 29 Interrupt Status                                                                                                           |
| 0x1110 | 273    | eTPU_GCR_CISR_ENG2[CIS30]                                                                                    | eTPU_B    | eTPU Engine B Channel 30 Interrupt Status                                                                                                           |
| 0x1120 | 274    | eTPU_GCR_CISR_ENG2[CIS31]                                                                                    | eTPU_B    | eTPU Engine B Channel 31 Interrupt Status                                                                                                           |
| 0x1130 | 275    | DSPIA_SR[TFUF]<br>DSPIA_SR[RFOF]<br>DSPIA_SR[SPEF]<br>DSPIA_SR[DPEF]                                         | DSPI_A    | DSPI_A combined overrun and parity error interrupt requests:<br><br>Transmit FIFO Underflow/Receive FIFO Overflow SPI Parity Error/DSI Parity Error |
| 0x1140 | 276    | DSPIA_SR[EOQF]                                                                                               | DSPI_A    | DSPI_A transmit FIFO End of Queue Flag                                                                                                              |
| 0x1150 | 277    | DSPIA_SR[TFFF]                                                                                               | DSPI_A    | DSPI_A Transmit FIFO Fill Flag                                                                                                                      |
| 0x1160 | 278    | DSPIA_SR[TCF]   DSPIA_SR[DDIF]                                                                               | DSPI_A    | DSPI_A Transfer Complete/DSI Data Match Flag                                                                                                        |
| 0x1170 | 279    | DSPIA_SR[RFDF]                                                                                               | DSPI_A    | DSPI_A Receive FIFO Drain Flag                                                                                                                      |
| 0x1180 | 280    | CANB_ESR1[BOFF_INT]  <br>[TWRNINT]   [RWRNINT]                                                               | FlexCAN_B | FlexCAN_B Bus Off, Transmit Warning, Receive Warning                                                                                                |
| 0x1190 | 281    | CANB_ESR1[ERR_INT]  <br>CANB_ERRSR[ECC_CE_INT]  <br>CANB_ERRSR[ECC_HANCE_INT]  <br>CANB_ERRSR[ECC_FANCE_INT] | FlexCAN_B | FlexCAN_B Error, FlexCAN_B ECC Correctable Error, FlexCAN_B ECC Host Access Non-Correctable Error, FlexCAN_B ECC CAN Access Non-Correctable Error   |
| 0x11A0 | 282    | Reserved                                                                                                     | FlexCAN_B | Reserved                                                                                                                                            |
| 0x11B0 | 283    | CANB_IFRL[BUF0]                                                                                              | FlexCAN_B | FlexCAN_B Buffer 0 Interrupt                                                                                                                        |
| 0x11C0 | 284    | CANB_IFRL[BUF1]                                                                                              | FlexCAN_B | FlexCAN_B Buffer 1 Interrupt                                                                                                                        |
| 0x11D0 | 285    | CANB_IFRL[BUF2]                                                                                              | FlexCAN_B | FlexCAN_B Buffer 2 Interrupt                                                                                                                        |
| 0x11E0 | 286    | CANB_IFRL[BUF3]                                                                                              | FlexCAN_B | FlexCAN_B Buffer 3 Interrupt                                                                                                                        |
| 0x11F0 | 287    | CANB_IFRL[BUF4]                                                                                              | FlexCAN_B | FlexCAN_B Buffer 4 Interrupt                                                                                                                        |
| 0x1200 | 288    | CANB_IFRL[BUF5]                                                                                              | FlexCAN_B | FlexCAN_B Buffer 5 Interrupt                                                                                                                        |
| 0x1210 | 289    | CANB_IFRL[BUF6]                                                                                              | FlexCAN_B | FlexCAN_B Buffer 6 Interrupt                                                                                                                        |
| 0x1220 | 290    | CANB_IFRL[BUF7]                                                                                              | FlexCAN_B | FlexCAN_B Buffer 7 Interrupt                                                                                                                        |

Table continues on the next page...

**Table 4-4. Interrupt vectors and request sources (continued)**

| Offset | Vector | Source                                                                                                       | Module    | Description                                                                                                                                                |
|--------|--------|--------------------------------------------------------------------------------------------------------------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x1230 | 291    | CANB_IFRL[BUF8]                                                                                              | FlexCAN_B | FlexCAN_B Buffer 8 Interrupt                                                                                                                               |
| 0x1240 | 292    | CANB_IFRL[BUF9]                                                                                              | FlexCAN_B | FlexCAN_B Buffer 9 Interrupt                                                                                                                               |
| 0x1250 | 293    | CANB_IFRL[BUF10]                                                                                             | FlexCAN_B | FlexCAN_B Buffer 10 Interrupt                                                                                                                              |
| 0x1260 | 294    | CANB_IFRL[BUF11]                                                                                             | FlexCAN_B | FlexCAN_B Buffer 11 Interrupt                                                                                                                              |
| 0x1270 | 295    | CANB_IFRL[BUF12]                                                                                             | FlexCAN_B | FlexCAN_B Buffer 12 Interrupt                                                                                                                              |
| 0x1280 | 296    | CANB_IFRL[BUF13]                                                                                             | FlexCAN_B | FlexCAN_B Buffer 13 Interrupt                                                                                                                              |
| 0x1290 | 297    | CANB_IFRL[BUF14]                                                                                             | FlexCAN_B | FlexCAN_B Buffer 14 Interrupt                                                                                                                              |
| 0x12A0 | 298    | CANB_IFRL[BUF15]                                                                                             | FlexCAN_B | FlexCAN_B Buffer 15 Interrupt                                                                                                                              |
| 0x12B0 | 299    | CANB_IFRL[BUF31:BUF16]                                                                                       | FlexCAN_B | FlexCAN_B Buffers 31-16 Interrupts                                                                                                                         |
| 0x12C0 | 300    | CANB_IFRH[BUF63:BUF32]                                                                                       | FlexCAN_B | FlexCAN_B Buffers 63-32 Interrupts                                                                                                                         |
| 0x12D0 | 301    | PIT_TFLG0[TIF]                                                                                               | PIT_RTI   | Periodic Interrupt Timer Interrupt 0                                                                                                                       |
| 0x12E0 | 302    | PIT_TFLG1[TIF]                                                                                               | PIT_RTI   | Periodic Interrupt Timer Interrupt 1                                                                                                                       |
| 0x12F0 | 303    | PIT_TFLG2[TIF]                                                                                               | PIT_RTI   | Periodic Interrupt Timer Interrupt 2                                                                                                                       |
| 0x1300 | 304    | PIT_TFLG3[TIF]                                                                                               | PIT_RTI   | Periodic Interrupt Timer Interrupt 3                                                                                                                       |
| 0x1310 | 305    | PIT_RTI_TFLG[TIF]                                                                                            | PIT_RTI   | Real Time Interrupt Interrupt                                                                                                                              |
| 0x1320 | 306    | Reserved                                                                                                     | —         | Reserved                                                                                                                                                   |
| 0x1330 | 307    | MCR[DONE]                                                                                                    | C55FMC    | Flash memory program/erase complete                                                                                                                        |
| 0x1340 | 308    | CAND_ESR1[BOFF_INT]  <br>[TWRNINT]   [RWRNINT]                                                               | FlexCAN_D | FlexCAN_D Bus Off, Transmit Warning,<br>Receive Warning                                                                                                    |
| 0x1350 | 309    | CAND_ESR1[ERR_INT]  <br>CAND_ERRSR[ECC_CE_INT]  <br>CAND_ERRSR[ECC_HANCE_INT]<br>  CAND_ERRSR[ECC_FANCE_INT] | FlexCAN_D | FlexCAN_D Error, FlexCAN_D ECC<br>Correctable Error, FlexCAN_D ECC Host<br>Access Non-Correctable Error, FlexCAN_D<br>ECC CAN Access Non-Correctable Error |
| 0x1360 | 310    | Reserved                                                                                                     | FlexCAN_D | Reserved                                                                                                                                                   |
| 0x1370 | 311    | CAND_IFRL[BUF0]                                                                                              | FlexCAN_D | FlexCAN_D buffer 0                                                                                                                                         |
| 0x1380 | 312    | CAND_IFRL[BUF1]                                                                                              | FlexCAN_D | FlexCAN_D buffer 1                                                                                                                                         |
| 0x1390 | 313    | CAND_IFRL[BUF2]                                                                                              | FlexCAN_D | FlexCAN_D buffer 2                                                                                                                                         |
| 0x13A0 | 314    | CAND_IFRL[BUF3]                                                                                              | FlexCAN_D | FlexCAN_D buffer 3                                                                                                                                         |
| 0x13B0 | 315    | CAND_IFRL[BUF4]                                                                                              | FlexCAN_D | FlexCAN_D buffer 4                                                                                                                                         |
| 0x13C0 | 316    | CAND_IFRL[BUF5]                                                                                              | FlexCAN_D | FlexCAN_D buffer 5                                                                                                                                         |
| 0x13D0 | 317    | CAND_IFRL[BUF6]                                                                                              | FlexCAN_D | FlexCAN_D buffer 6                                                                                                                                         |
| 0x13E0 | 318    | CAND_IFRL[BUF7]                                                                                              | FlexCAN_D | FlexCAN_D buffer 7                                                                                                                                         |
| 0x13F0 | 319    | CAND_IFRL[BUF8]                                                                                              | FlexCAN_D | FlexCAN_D buffer 8                                                                                                                                         |
| 0x1400 | 320    | CAND_IFRL[BUF9]                                                                                              | FlexCAN_D | FlexCAN_D buffer 9                                                                                                                                         |
| 0x1410 | 321    | CAND_IFRL[BUF10]                                                                                             | FlexCAN_D | FlexCAN_D buffer 10                                                                                                                                        |
| 0x1420 | 322    | CAND_IFRL[BUF11]                                                                                             | FlexCAN_D | FlexCAN_D buffer 11                                                                                                                                        |
| 0x1430 | 323    | CAND_IFRL[BUF12]                                                                                             | FlexCAN_D | FlexCAN_D buffer 12                                                                                                                                        |
| 0x1440 | 324    | CAND_IFRL[BUF13]                                                                                             | FlexCAN_D | FlexCAN_D buffer 13                                                                                                                                        |
| 0x1450 | 325    | CAND_IFRL[BUF14]                                                                                             | FlexCAN_D | FlexCAN_D buffer 14                                                                                                                                        |
| 0x1460 | 326    | CAND_IFRL[BUF15]                                                                                             | FlexCAN_D | FlexCAN_D buffer 15                                                                                                                                        |
| 0x1470 | 327    | CAND_IFRL[BUF31:BUF16]                                                                                       | FlexCAN_D | FlexCAN_D buffer 16-31                                                                                                                                     |

Table continues on the next page...

**Table 4-4. Interrupt vectors and request sources (continued)**

| Offset | Vector | Source                                                                                                                                                                                                                         | Module    | Description              |
|--------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|--------------------------|
| 0x1480 | 328    | CAND_IFRH[BUF63:BUF32]                                                                                                                                                                                                         | FlexCAN_D | FlexCAN_D buffer 32-63   |
| 0x1490 | 329    | SENT_0_GBL_STATUS[FMDU]  <br>SENT_0_GBL_STATUS[SMDU]  <br>SENT_0_GBL_STATUS[FMFO]   all<br>sources of vectors 330-335                                                                                                          | SRX_0     | SENT_0 Module Interrupts |
| 0x14A0 | 330    | SENT_0_CH0_STATUS[CAL_RES<br>YNCI CAL_20_50ISMSG_OFLWI<br>FMSG_OFLWI NUM_EDGES_ERRI<br>FMSG_CRC_ERRI<br>SMSG_CRC_ERRINIB_VAL_ERRI<br>CAL_DIAG_ERRICAL_LEN_ERRI<br>PP_DIAG_ERR]  <br>SENT_0_FMSG_RDY[0]  <br>SENT_0_SMSG_RDY[0] | SRX_0     | SENT_0_CH0 Interrupts    |
| 0x14B0 | 331    | SENT_0_CH1_STATUS[CAL_RES<br>YNCI CAL_20_50ISMSG_OFLWI<br>FMSG_OFLWI NUM_EDGES_ERRI<br>FMSG_CRC_ERRI<br>SMSG_CRC_ERRINIB_VAL_ERRI<br>CAL_DIAG_ERRICAL_LEN_ERRI<br>PP_DIAG_ERR]  <br>SENT_0_FMSG_RDY[1]  <br>SENT_0_SMSG_RDY[1] | SRX_0     | SENT_0_CH1 Interrupts    |
| 0x14C0 | 332    | SENT_0_CH2_STATUS[CAL_RES<br>YNCI CAL_20_50ISMSG_OFLWI<br>FMSG_OFLWI NUM_EDGES_ERRI<br>FMSG_CRC_ERRI<br>SMSG_CRC_ERRINIB_VAL_ERRI<br>CAL_DIAG_ERRICAL_LEN_ERRI<br>PP_DIAG_ERR]  <br>SENT_0_FMSG_RDY[2]  <br>SENT_0_SMSG_RDY[2] | SRX_0     | SENT_0_CH2 Interrupts    |
| 0x14D0 | 333    | SENT_0_CH3_STATUS[CAL_RES<br>YNCI CAL_20_50ISMSG_OFLWI<br>FMSG_OFLWI NUM_EDGES_ERRI<br>FMSG_CRC_ERRI<br>SMSG_CRC_ERRINIB_VAL_ERRI<br>CAL_DIAG_ERRICAL_LEN_ERRI<br>PP_DIAG_ERR]  <br>SENT_0_FMSG_RDY[3]  <br>SENT_0_SMSG_RDY[3] | SRX_0     | SENT_0_CH3 Interrupts    |
| 0x14E0 | 334    | SENT_0_CH4_STATUS[CAL_RES<br>YNCI CAL_20_50ISMSG_OFLWI<br>FMSG_OFLWI NUM_EDGES_ERRI<br>FMSG_CRC_ERRI<br>SMSG_CRC_ERRINIB_VAL_ERRI<br>CAL_DIAG_ERRICAL_LEN_ERRI<br>PP_DIAG_ERR]  <br>SENT_0_FMSG_RDY[4]  <br>SENT_0_SMSG_RDY[4] | SRX_0     | SENT_0_CH4 Interrupts    |

*Table continues on the next page...*

**Table 4-4. Interrupt vectors and request sources (continued)**

| Offset | Vector | Source                                                                                                                                                                                                                         | Module | Description              |
|--------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|--------------------------|
| 0x14F0 | 335    | SENT_0_CH5_STATUS[CAL_RES<br>YNCI CAL_20_50ISMSG_OFLWI<br>FMSG_OFLWI NUM_EDGES_ERRI<br>FMSG_CRC_ERRI<br>SMSG_CRC_ERRINIB_VAL_ERRI<br>CAL_DIAG_ERRICAL_LEN_ERRI<br>PP_DIAG_ERR]  <br>SENT_0_FMSG_RDY[5]  <br>SENT_0_SMSG_RDY[5] | SRX_0  | SENT_0_CH5 Interrupts    |
| 0x1500 | 336    | SENT_1_GBL_STATUS[FMDU]  <br>SENT_1_GBL_STATUS[SMDU]  <br>SENT_1_GBL_STATUS[FMFO]   all<br>sources of vectors 337-342                                                                                                          | SRX_1  | SENT_1 Module Interrupts |
| 0x1510 | 337    | SENT_1_CH0_STATUS[CAL_RES<br>YNCI CAL_20_50ISMSG_OFLWI<br>FMSG_OFLWI NUM_EDGES_ERRI<br>FMSG_CRC_ERRI<br>SMSG_CRC_ERRINIB_VAL_ERRI<br>CAL_DIAG_ERRICAL_LEN_ERRI<br>PP_DIAG_ERR]  <br>SENT_1_FMSG_RDY[0]  <br>SENT_1_SMSG_RDY[0] | SRX_1  | SENT_1_CH0 Interrupts    |
| 0x1520 | 338    | SENT_1_CH1_STATUS[CAL_RES<br>YNCI CAL_20_50ISMSG_OFLWI<br>FMSG_OFLWI NUM_EDGES_ERRI<br>FMSG_CRC_ERRI<br>SMSG_CRC_ERRINIB_VAL_ERRI<br>CAL_DIAG_ERRICAL_LEN_ERRI<br>PP_DIAG_ERR]  <br>SENT_1_FMSG_RDY[1]  <br>SENT_1_SMSG_RDY[1] | SRX_1  | SENT_1_CH1 Interrupts    |
| 0x1530 | 339    | SENT_1_CH2_STATUS[CAL_RES<br>YNCI CAL_20_50ISMSG_OFLWI<br>FMSG_OFLWI NUM_EDGES_ERRI<br>FMSG_CRC_ERRI<br>SMSG_CRC_ERRINIB_VAL_ERRI<br>CAL_DIAG_ERRICAL_LEN_ERRI<br>PP_DIAG_ERR]  <br>SENT_1_FMSG_RDY[2]  <br>SENT_1_SMSG_RDY[2] | SRX_1  | SENT_1_CH2 Interrupts    |
| 0x1540 | 340    | SENT_1_CH3_STATUS[CAL_RES<br>YNCI CAL_20_50ISMSG_OFLWI<br>FMSG_OFLWI NUM_EDGES_ERRI<br>FMSG_CRC_ERRI<br>SMSG_CRC_ERRINIB_VAL_ERRI<br>CAL_DIAG_ERRICAL_LEN_ERRI<br>PP_DIAG_ERR]  <br>SENT_1_FMSG_RDY[3]  <br>SENT_1_SMSG_RDY[3] | SRX_1  | SENT_1_CH3 Interrupts    |
| 0x1550 | 341    | SENT_1_CH4_STATUS[CAL_RES<br>YNCI CAL_20_50ISMSG_OFLWI<br>FMSG_OFLWI NUM_EDGES_ERRI<br>FMSG_CRC_ERRI                                                                                                                           | SRX_1  | SENT_1_CH4 Interrupts    |

Table continues on the next page...

**Table 4-4. Interrupt vectors and request sources (continued)**

| Offset | Vector | Source                                                                                                                                                                                                                         | Module   | Description                                                   |
|--------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------------------------------------------------------------|
|        |        | SMSG_CRC_ERRINIB_VAL_ERRI<br>CAL_DIAG_ERRICAL_LEN_ERRI<br>PP_DIAG_ERR]  <br>SENT_1_FMSG_RDY[4]  <br>SENT_1_SMSG_RDY[4]                                                                                                         |          |                                                               |
| 0x1560 | 342    | SENT_1_CH5_STATUS[CAL_RES<br>YNCI CAL_20_50ISMSG_OFLWI<br>FMSG_OFLWI NUM_EDGES_ERRI<br>FMSG_CRC_ERRI<br>SMSG_CRC_ERRINIB_VAL_ERRI<br>CAL_DIAG_ERRICAL_LEN_ERRI<br>PP_DIAG_ERR]  <br>SENT_1_FMSG_RDY[5]  <br>SENT_1_SMSG_RDY[5] | SRX_1    | SENT_1_CH5 Interrupts                                         |
| 0x1570 | 343    | HVD_FLASH, HVD_HV,<br>HVD_CORE, LVD_CORE_COLD of<br>PMC_LVD_HVD_EVENT_STATUS                                                                                                                                                   | PMC      | Power Management Controller Interrupts                        |
| 0x1580 | 344    | TEMP0_0, TEMP0_2, TEMP0_3,<br>TEMP1_0, TEMP1_2, TEMP1_3 of<br>PMC_ESR_TD                                                                                                                                                       | PMC Temp | Temperature Sensor Interrupts                                 |
| 0x1590 | 345    | JDC_MSR[JIN_INT]  <br>JDC_MSR[JOUT_INT]                                                                                                                                                                                        | JDC      | JDC Interrupts                                                |
| 0x15A0 | 346    | Reserved                                                                                                                                                                                                                       | –        | Reserved                                                      |
| 0x15B0 | 347    | Reserved                                                                                                                                                                                                                       | –        | Reserved                                                      |
| 0x15C0 | 348    | m_can0_int0   m_can0_int1  <br>m_can1_int0   m_can1_int1                                                                                                                                                                       | M_CAN    | M_CAN0_0, M_CAN0_1, M_CAN1_0,<br>M_CAN1_1 Combined Interrupts |
| 0x15D0 | 349    | Single bit Correction   Multi bit<br>Detection                                                                                                                                                                                 | ERM      | ERM Combined Interrupts                                       |
| 0x15E0 | 350    | CMU_0_ISR[FHHIIFLLIOLRI]  <br>CMU_1_ISR[FHHIIFLLI]                                                                                                                                                                             | CMU      | CMU_0, CMU_1 Clock Error Interrupts                           |
| 0x15F0 | 351    | CMU_2_ISR[FHHIIFLLI]  <br>CMU_3_ISR[FHHIIFLLI]                                                                                                                                                                                 | CMU      | CMU_2, CMU_3 Clock Error Interrupts                           |
| 0x1600 | 352    | CMU_4_ISR[FHHIIFLLI]  <br>CMU_5_ISR[FHHIIFLLI]                                                                                                                                                                                 | CMU      | CMU_4, CMU_5 Clock Error Interrupts                           |
| 0x1610 | 353    | CMU_6_ISR[FHHIIFLLI]  <br>CMU_7_ISR[FHHIIFLLI]                                                                                                                                                                                 | CMU      | CMU_6, CMU_7 Clock Error Interrupts                           |
| 0x1620 | 354    | CMU_8_ISR[FHHIIFLLI]                                                                                                                                                                                                           | CMU      | CMU_8 Clock Error Interrupts                                  |
| 0x1630 | 355    | Reserved                                                                                                                                                                                                                       | CMU      | Reserved                                                      |
| 0x1640 | 356    | Reserved                                                                                                                                                                                                                       | CMU      | Reserved                                                      |
| 0x1650 | 357    | Reserved                                                                                                                                                                                                                       | CMU      | Reserved                                                      |
| 0x0C50 | 358    | Reserved                                                                                                                                                                                                                       | –        | Reserved                                                      |
| 0x0C50 | 359    | Reserved                                                                                                                                                                                                                       | –        | Reserved                                                      |
| 0x0C50 | 360    | Reserved                                                                                                                                                                                                                       | –        | Reserved                                                      |
| 0x0C50 | 361    | Reserved                                                                                                                                                                                                                       | –        | Reserved                                                      |
| 0x0C50 | 362    | Reserved                                                                                                                                                                                                                       | –        | Reserved                                                      |
| 0x0C50 | 363    | Reserved                                                                                                                                                                                                                       | –        | Reserved                                                      |

Table continues on the next page...

**Table 4-4. Interrupt vectors and request sources (continued)**

| Offset | Vector | Source                                                                                                | Module  | Description                                                                                                                                |
|--------|--------|-------------------------------------------------------------------------------------------------------|---------|--------------------------------------------------------------------------------------------------------------------------------------------|
| 0x0C50 | 364    | Reserved                                                                                              | –       | Reserved                                                                                                                                   |
| 0x0C50 | 365    | Reserved                                                                                              | –       | Reserved                                                                                                                                   |
| 0x0C50 | 366    | Reserved                                                                                              | –       | Reserved                                                                                                                                   |
| 0x0C50 | 367    | Reserved                                                                                              | –       | Reserved                                                                                                                                   |
| 0x0C50 | 368    | Reserved                                                                                              | –       | Reserved                                                                                                                                   |
| 0x1710 | 369    | eTPU_SCR_MCR[SDMERR,SSAERR,SCMERR,WDT01,MGE1,ILF1,LTO1,SCMMISF] and eTPU_MESR_MESR[DCERR,CCERR]       | eTPU_C  | eTPU Engine C Global Exception                                                                                                             |
| 0x1720 | 370    | eTPU_GCR_CISR_ENG1[CIS0]                                                                              | eTPU_C  | eTPU Engine C Channel 0 Interrupt Status                                                                                                   |
| 0x1730 | 371    | eTPU_GCR_CISR_ENG1[CIS1]                                                                              | eTPU_C  | eTPU Engine C Channel 1 Interrupt Status                                                                                                   |
| 0x1740 | 372    | eTPU_GCR_CISR_ENG1[CIS2]                                                                              | eTPU_C  | eTPU Engine C Channel 2 Interrupt Status                                                                                                   |
| 0x1750 | 373    | eTPU_GCR_CISR_ENG1[CIS3]                                                                              | eTPU_C  | eTPU Engine C Channel 3 Interrupt Status                                                                                                   |
| 0x1760 | 374    | eTPU_GCR_CISR_ENG1[CIS4]                                                                              | eTPU_C  | eTPU Engine C Channel 4 Interrupt Status                                                                                                   |
| 0x1770 | 375    | eTPU_GCR_CISR_ENG1[CIS5]                                                                              | eTPU_C  | eTPU Engine C Channel 5 Interrupt Status                                                                                                   |
| 0x1780 | 376    | eTPU_GCR_CISR_ENG1[CIS6]                                                                              | eTPU_C  | eTPU Engine C Channel 6 Interrupt Status                                                                                                   |
| 0x1790 | 377    | eTPU_GCR_CISR_ENG1[CIS7]                                                                              | eTPU_C  | eTPU Engine C Channel 7 Interrupt Status                                                                                                   |
| 0x17A0 | 378    | eTPU_GCR_CISR_ENG1[CIS8]                                                                              | eTPU_C  | eTPU Engine C Channel 8 Interrupt Status                                                                                                   |
| 0x17B0 | 379    | eTPU_GCR_CISR_ENG1[CIS9]                                                                              | eTPU_C  | eTPU Engine C Channel 9 Interrupt Status                                                                                                   |
| 0x17C0 | 380    | eTPU_GCR_CISR_ENG1[CIS10]                                                                             | eTPU_C  | eTPU Engine C Channel 10 Interrupt Status                                                                                                  |
| 0x17D0 | 381    | eTPU_GCR_CISR_ENG1[CIS11]                                                                             | eTPU_C  | eTPU Engine C Channel 11 Interrupt Status                                                                                                  |
| 0x17E0 | 382    | eTPU_GCR_CISR_ENG1[CIS12]                                                                             | eTPU_C  | eTPU Engine C Channel 12 Interrupt Status                                                                                                  |
| 0x17F0 | 383    | eTPU_GCR_CISR_ENG1[CIS13]                                                                             | eTPU_C  | eTPU Engine C Channel 13 Interrupt Status                                                                                                  |
| 0x1800 | 384    | eTPU_GCR_CISR_ENG1[CIS14]                                                                             | eTPU_C  | eTPU Engine C Channel 14 Interrupt Status                                                                                                  |
| 0x1810 | 385    | eTPU_GCR_CISR_ENG1[CIS15]                                                                             | eTPU_C  | eTPU Engine C Channel 15 Interrupt Status                                                                                                  |
| 0x1820 | 386    | eTPU_GCR_CISR_ENG1[CIS16]                                                                             | eTPU_C  | eTPU Engine C Channel 16 Interrupt Status                                                                                                  |
| 0x1830 | 387    | eTPU_GCR_CISR_ENG1[CIS17]                                                                             | eTPU_C  | eTPU Engine C Channel 17 Interrupt Status                                                                                                  |
| 0x1840 | 388    | eTPU_GCR_CISR_ENG1[CIS18]                                                                             | eTPU_C  | eTPU Engine C Channel 18 Interrupt Status                                                                                                  |
| 0x1850 | 389    | eTPU_GCR_CISR_ENG1[CIS19]                                                                             | eTPU_C  | eTPU Engine C Channel 19 Interrupt Status                                                                                                  |
| 0x1860 | 390    | eTPU_GCR_CISR_ENG1[CIS20]                                                                             | eTPU_C  | eTPU Engine C Channel 20 Interrupt Status                                                                                                  |
| 0x1870 | 391    | eTPU_GCR_CISR_ENG1[CIS21]                                                                             | eTPU_C  | eTPU Engine C Channel 21 Interrupt Status                                                                                                  |
| 0x1880 | 392    | eTPU_GCR_CISR_ENG1[CIS22]                                                                             | eTPU_C  | eTPU Engine C Channel 22 Interrupt Status                                                                                                  |
| 0x1890 | 393    | eTPU_GCR_CISR_ENG1[CIS23]                                                                             | eTPU_C  | eTPU Engine C Channel 23 Interrupt Status                                                                                                  |
| 0x18A0 | 394    | EQADCB_FISR <sub>x</sub> [TORF]<br>EQADCB_FISR <sub>x</sub> [RFOF]<br>EQADCB_FISR <sub>x</sub> [CFUF] | eQADC_B | eQADC combined overrun interrupt requests from all of the FIFOs:<br><br>Trigger Overrun, Receive FIFO Overflow, and command FIFO Underflow |
| 0x18B0 | 395    | EQADCB_FISR0[NCF]                                                                                     | eQADC_B | eQADC command FIFO 0 Non-Coherency Flag                                                                                                    |
| 0x18C0 | 396    | EQADCB_FISR0[PF]                                                                                      | eQADC_B | eQADC command FIFO 0 Pause Flag                                                                                                            |

Table continues on the next page...

**Table 4-4. Interrupt vectors and request sources (continued)**

| Offset | Vector | Source                 | Module  | Description                                          |
|--------|--------|------------------------|---------|------------------------------------------------------|
| 0x18D0 | 397    | EQADCB_FISR0[EOQF]     | eQADC_B | eQADC command FIFO 0 command queue End of Queue Flag |
| 0x18E0 | 398    | EQADCB_FISR0[CFFF]     | eQADC_B | eQADC Command FIFO 0 Fill Flag                       |
| 0x18F0 | 399    | EQADCB_FISR0[RFDF]     | eQADC_B | eQADC Receive FIFO 0 Drain Flag                      |
| 0x1900 | 400    | EQADCB_FISR1[NCF]      | eQADC_B | eQADC command FIFO 1 Non-Coherency Flag              |
| 0x1910 | 401    | EQADCB_FISR1[PF]       | eQADC_B | eQADC command FIFO 1 Pause Flag                      |
| 0x1920 | 402    | EQADCB_FISR1[EOQF]     | eQADC_B | eQADC command FIFO 1 command queue End of Queue Flag |
| 0x1930 | 403    | EQADCB_FISR1[CFFF]     | eQADC_B | eQADC Command FIFO 1 Fill Flag                       |
| 0x1940 | 404    | EQADCB_FISR1[RFDF]     | eQADC_B | eQADC Receive FIFO 1 Drain Flag                      |
| 0x1950 | 405    | EQADCB_FISR2[NCF]      | eQADC_B | eQADC command FIFO 2 Non-Coherency Flag              |
| 0x1960 | 406    | EQADCB_FISR2[PF]       | eQADC_B | eQADC command FIFO 2 Pause Flag                      |
| 0x1970 | 407    | EQADCB_FISR2[EOQF]     | eQADC_B | eQADC command FIFO 2 command queue End of Queue Flag |
| 0x1980 | 408    | EQADCB_FISR2[CFFF]     | eQADC_B | eQADC Command FIFO 2 Fill Flag                       |
| 0x1990 | 409    | EQADCB_FISR2[RFDF]     | eQADC_B | eQADC Receive FIFO 2 Drain Flag                      |
| 0x19A0 | 410    | EQADCB_FISR3[NCF]      | eQADC_B | eQADC command FIFO 3 Non-Coherency Flag              |
| 0x19B0 | 411    | EQADCB_FISR3[PF]       | eQADC_B | eQADC command FIFO 3 Pause Flag                      |
| 0x19C0 | 412    | EQADCB_FISR3[EOQF]     | eQADC_B | eQADC command FIFO 3 command queue End of Queue Flag |
| 0x19D0 | 413    | EQADCB_FISR3[CFFF]     | eQADC_B | eQADC Command FIFO 3 Fill Flag                       |
| 0x19E0 | 414    | EQADCB_FISR3[RFDF]     | eQADC_B | eQADC Receive FIFO 3 Drain Flag                      |
| 0x19F0 | 415    | EQADCB_FISR4[NCF]      | eQADC_B | eQADC command FIFO 4 Non-Coherency Flag              |
| 0x1A00 | 416    | EQADCB_FISR4[PF]       | eQADC_B | eQADC command FIFO 4 Pause Flag                      |
| 0x1A10 | 417    | EQADCB_FISR4[EOQF]     | eQADC_B | eQADC command FIFO 4 command queue End of Queue Flag |
| 0x1A20 | 418    | EQADCB_FISR4[CFFF]     | eQADC_B | eQADC Command FIFO 4 Fill Flag                       |
| 0x1A30 | 419    | EQADCB_FISR4[RFDF]     | eQADC_B | eQADC Receive FIFO 4 Drain Flag                      |
| 0x1A40 | 420    | EQADCB_FISR5[NCF]      | eQADC_B | eQADC command FIFO 5 Non-Coherency Flag              |
| 0x1A50 | 421    | EQADCB_FISR5[PF]       | eQADC_B | eQADC command FIFO 5 Pause Flag                      |
| 0x1A60 | 422    | EQADCB_FISR5[EOQF]     | eQADC_B | eQADC command FIFO 5 command queue End of Queue Flag |
| 0x1A70 | 423    | EQADCB_FISR5[CFFF]     | eQADC_B | eQADC Command FIFO 5 Fill Flag                       |
| 0x1A80 | 424    | EQADCB_FISR5[RFDF]     | eQADC_B | eQADC Receive FIFO 5 Drain Flag                      |
| 0x1A90 | 425    | EDMAB_ERRH[ERR31:ERR0] | eDMA_B  | eDMA channel Error flags 0-31                        |
| 0x1AA0 | 426    | EDMAB_IRQRH[INT0]      | eDMA_B  | eDMA channel Interrupt 0                             |
| 0x1AB0 | 427    | EDMAB_IRQRH[INT1]      | eDMA_B  | eDMA channel Interrupt 1                             |
| 0x1AC0 | 428    | EDMAB_IRQRH[INT2]      | eDMA_B  | eDMA channel Interrupt 2                             |

Table continues on the next page...

**Table 4-4. Interrupt vectors and request sources (continued)**

| Offset | Vector | Source                                | Module  | Description                 |
|--------|--------|---------------------------------------|---------|-----------------------------|
| 0x1AD0 | 429    | EDMAB_IRQRH[INT3]                     | eDMA_B  | eDMA channel Interrupt 3    |
| 0x1AE0 | 430    | EDMAB_IRQRH[INT4]                     | eDMA_B  | eDMA channel Interrupt 4    |
| 0x1AF0 | 431    | EDMAB_IRQRH[INT5]                     | eDMA_B  | eDMA channel Interrupt 5    |
| 0x1B00 | 432    | EDMAB_IRQRH[INT6]                     | eDMA_B  | eDMA channel Interrupt 6    |
| 0x1B10 | 433    | EDMAB_IRQRH[INT7]                     | eDMA_B  | eDMA channel Interrupt 7    |
| 0x1B20 | 434    | EDMAB_IRQRH[INT8]                     | eDMA_B  | eDMA channel Interrupt 8    |
| 0x1B30 | 435    | EDMAB_IRQRH[INT9]                     | eDMA_B  | eDMA channel Interrupt 9    |
| 0x1B40 | 436    | EDMAB_IRQRH[INT10]                    | eDMA_B  | eDMA channel Interrupt 10   |
| 0x1B50 | 437    | EDMAB_IRQRH[INT11]                    | eDMA_B  | eDMA channel Interrupt 11   |
| 0x1B60 | 438    | EDMAB_IRQRH[INT12]                    | eDMA_B  | eDMA channel Interrupt 12   |
| 0x1B70 | 439    | EDMAB_IRQRH[INT13]                    | eDMA_B  | eDMA channel Interrupt 13   |
| 0x1B80 | 440    | EDMAB_IRQRH[INT14]                    | eDMA_B  | eDMA channel Interrupt 14   |
| 0x1B90 | 441    | EDMAB_IRQRH[INT15]                    | eDMA_B  | eDMA channel Interrupt 15   |
| 0x1BA0 | 442    | EDMAB_IRQRH[INT16]                    | eDMA_B  | eDMA channel Interrupt 16   |
| 0x1BB0 | 443    | EDMAB_IRQRH[INT17]                    | eDMA_B  | eDMA channel Interrupt 17   |
| 0x1BC0 | 444    | EDMAB_IRQRH[INT18]                    | eDMA_B  | eDMA channel Interrupt 18   |
| 0x1BD0 | 445    | EDMAB_IRQRH[INT19]                    | eDMA_B  | eDMA channel Interrupt 19   |
| 0x1BE0 | 446    | EDMAB_IRQRH[INT20]                    | eDMA_B  | eDMA channel Interrupt 20   |
| 0x1BF0 | 447    | EDMAB_IRQRH[INT21]                    | eDMA_B  | eDMA channel Interrupt 21   |
| 0x1C00 | 448    | EDMAB_IRQRH[INT22]                    | eDMA_B  | eDMA channel Interrupt 22   |
| 0x1C10 | 449    | EDMAB_IRQRH[INT23]                    | eDMA_B  | eDMA channel Interrupt 23   |
| 0x1C20 | 450    | EDMAB_IRQRH[INT24]                    | eDMA_B  | eDMA channel Interrupt 24   |
| 0x1C30 | 451    | EDMAB_IRQRH[INT25]                    | eDMA_B  | eDMA channel Interrupt 25   |
| 0x1C40 | 452    | EDMAB_IRQRH[INT26]                    | eDMA_B  | eDMA channel Interrupt 26   |
| 0x1C50 | 453    | EDMAB_IRQRH[INT27]                    | eDMA_B  | eDMA channel Interrupt 27   |
| 0x1C60 | 454    | EDMAB_IRQRH[INT28]                    | eDMA_B  | eDMA channel Interrupt 28   |
| 0x1C70 | 455    | EDMAB_IRQRH[INT29]                    | eDMA_B  | eDMA channel Interrupt 29   |
| 0x1C80 | 456    | EDMAB_IRQRH[INT30]                    | eDMA_B  | eDMA channel Interrupt 30   |
| 0x1C90 | 457    | EDMAB_IRQRH[INT31]                    | eDMA_B  | eDMA channel Interrupt 31   |
| 0x1CA0 | 458    | SD_ADC1   SD_ADC2   SD_ADC3   SD_ADC4 | SDADC   | SDADC1 to SDADC4 Interrupts |
| 0x1CB0 | 459    | eMIOS_1_GFLAG[F16]                    | eMIOS_1 | eMIOS_1 channel 16 Flag     |
| 0x1CC0 | 460    | eMIOS_1_GFLAG[F17]                    | eMIOS_1 | eMIOS_1 channel 17 Flag     |
| 0x1CD0 | 461    | eMIOS_1_GFLAG[F18]                    | eMIOS_1 | eMIOS_1 channel 18 Flag     |
| 0x1CE0 | 462    | eMIOS_1_GFLAG[F19]                    | eMIOS_1 | eMIOS_1 channel 19 Flag     |
| 0x1CF0 | 463    | eMIOS_1_GFLAG[F20]                    | eMIOS_1 | eMIOS_1 channel 20 Flag     |
| 0x1D00 | 464    | eMIOS_1_GFLAG[F21]                    | eMIOS_1 | eMIOS_1 channel 21 Flag     |
| 0x1D10 | 465    | eMIOS_1_GFLAG[F22]                    | eMIOS_1 | eMIOS_1 channel 22 Flag     |
| 0x1D20 | 466    | eMIOS_1_GFLAG[F23]                    | eMIOS_1 | eMIOS_1 channel 23 Flag     |
| 0x1D30 | 467    | Reserved                              | —       | Reserved                    |

Table continues on the next page...



**Table 4-4. Interrupt vectors and request sources (continued)**

| Offset | Vector | Source                                                                                                                                                                                                                                                                                                                    | Module | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|--------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x1D40 | 468    | Reserved                                                                                                                                                                                                                                                                                                                  | —      | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 0x1D50 | 469    | Reserved                                                                                                                                                                                                                                                                                                                  | —      | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 0x1D60 | 470    | Reserved                                                                                                                                                                                                                                                                                                                  | —      | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 0x1D70 | 471    | Reserved                                                                                                                                                                                                                                                                                                                  | —      | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 0x1D80 | 472    | Reserved                                                                                                                                                                                                                                                                                                                  | —      | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 0x1D90 | 473    | ESCIC_SR[TDRE]<br>ESCIC_SR[TC]<br>ESCIC_SR[RDRF]<br>ESCIC_SR[IDLE]<br>ESCIC_SR[OR]<br>ESCIC_SR[NF]<br>ESCIC_SR[FE]<br>ESCIC_SR[PF]<br>ESCIC_SR[BERR]<br>ESCIC_SR[RXRDY]<br>ESCIC_SR[TXRDY]<br>ESCIC_SR[LWAKE]<br>ESCIC_SR[STO]<br>ESCIC_SR[PBERR]<br>ESCIC_SR[CERR]<br>ESCIC_SR[CKERR]<br>ESCIC_SR[FRC]<br>ESCIC_SR[OVFL] | eSCI_C | Combined Interrupt Requests of ESCI Module C:<br><br>Transmit Data Register Empty, Transmit Complete, Receive Data Register Full, Idle line, Overrun, Noise Flag, Framing Error Flag, and Parity Error Flag interrupt requests, SCI Status Register 2 Bit Error interrupt request, LIN Status Register 1 Receive Data Ready, Transmit Data Ready, Received LIN Wakeup Signal, Slave TimeOut, Physical Bus Error, CRC Error, Checksum Error, Frame Complete interrupts requests, and LIN Status Register 2 Receive Register Overflow |
| 0x1DA0 | 474    | ESCID_SR[TDRE]<br>ESCID_SR[TC]<br>ESCID_SR[RDRF]<br>ESCID_SR[IDLE]<br>ESCID_SR[OR]<br>ESCID_SR[NF]<br>ESCID_SR[FE]<br>ESCID_SR[PF]<br>ESCID_SR[BERR]<br>ESCID_SR[RXRDY]<br>ESCID_SR[TXRDY]<br>ESCID_SR[LWAKE]<br>ESCID_SR[STO]<br>ESCID_SR[PBERR]<br>ESCID_SR[CERR]                                                       | eSCI_D | Combined Interrupt Requests of ESCI Module D:<br><br>Transmit Data Register Empty, Transmit Complete, Receive Data Register Full, Idle line, Overrun, Noise Flag, Framing Error Flag, and Parity Error Flag interrupt requests, SCI Status Register 2 Bit Error interrupt request, LIN Status Register 1 Receive Data Ready, Transmit Data Ready, Received LIN Wakeup Signal, Slave TimeOut, Physical Bus Error, CRC Error, Checksum Error, Frame Complete interrupts requests, and LIN Status Register 2 Receive Register Overflow |

Table continues on the next page...

**Table 4-4. Interrupt vectors and request sources (continued)**

| Offset | Vector | Source                                                                                                                                                                                                                                                                                                                    | Module | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|--------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|        |        | ESCID_SR[CKERR]<br>ESCID_SR[FRC]<br>ESCID_SR[OVFL]                                                                                                                                                                                                                                                                        |        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 0x1DB0 | 475    | ESCIE_SR[TDRE]<br>ESCIE_SR[TC]<br>ESCIE_SR[RDRF]<br>ESCIE_SR[IDLE]<br>ESCIE_SR[OR]<br>ESCIE_SR[NF]<br>ESCIE_SR[FE]<br>ESCIE_SR[PF]<br>ESCIE_SR[BERR]<br>ESCIE_SR[RXRDY]<br>ESCIE_SR[TXRDY]<br>ESCIE_SR[LWAKE]<br>ESCIE_SR[STO]<br>ESCIE_SR[PBERR]<br>ESCIE_SR[CERR]<br>ESCIE_SR[CKERR]<br>ESCIE_SR[FRC]<br>ESCIE_SR[OVFL] | eSCI_E | Combined Interrupt Requests of ESCI Module E:<br><br>Transmit Data Register Empty, Transmit Complete, Receive Data Register Full, Idle line, Overrun, Noise Flag, Framing Error Flag, and Parity Error Flag interrupt requests, SCI Status Register 2 Bit Error interrupt request, LIN Status Register 1 Receive Data Ready, Transmit Data Ready, Received LIN Wakeup Signal, Slave TimeOut, Physical Bus Error, CRC Error, Checksum Error, Frame Complete interrupts requests, and LIN Status Register 2 Receive Register Overflow |
| 0x1DC0 | 476    | Reserved                                                                                                                                                                                                                                                                                                                  | —      | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 0x1DD0 | 477    | Reserved                                                                                                                                                                                                                                                                                                                  | —      | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 0x1DE0 | 478    | Reserved                                                                                                                                                                                                                                                                                                                  | —      | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 0x1DF0 | 479    | Reserved                                                                                                                                                                                                                                                                                                                  | —      | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 0x1E00 | 480    | Reserved                                                                                                                                                                                                                                                                                                                  | —      | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 0x1E10 | 481    | Reserved                                                                                                                                                                                                                                                                                                                  | —      | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 0x1E20 | 482    | Reserved                                                                                                                                                                                                                                                                                                                  | —      | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 0x1E30 | 483    | Reserved                                                                                                                                                                                                                                                                                                                  | —      | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 0x1E40 | 484    | Reserved                                                                                                                                                                                                                                                                                                                  |        | Reserved for DEC_M                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 0x1E50 | 485    | Reserved                                                                                                                                                                                                                                                                                                                  |        | Reserved for DEC_N                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 0x1E60 | 486    | Reserved                                                                                                                                                                                                                                                                                                                  |        | Reserved for DEC_O                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 0x1E70 | 487    | Reserved                                                                                                                                                                                                                                                                                                                  |        | Reserved for DEC_P                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 0x1E80 | 488    | EDMAB_ERRH[ERR63:ERR32]                                                                                                                                                                                                                                                                                                   | eDMA_B | eDMA channel Error flags 32-63                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 0x1E90 | 489    | EDMAB_IRQRH[INT32:39]                                                                                                                                                                                                                                                                                                     | eDMA_B | eDMA channel Interrupts 32-39                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 0x1EA0 | 490    | EDMAB_IRQRH[INT40:47]                                                                                                                                                                                                                                                                                                     | eDMA_B | eDMA channel Interrupts 40-47                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 0x1EB0 | 491    | EDMAB_IRQRH[INT48:55]                                                                                                                                                                                                                                                                                                     | eDMA_B | eDMA channel Interrupts 48-55                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 0x1EC0 | 492    | EDMAB_IRQRH[INT56:63]                                                                                                                                                                                                                                                                                                     | eDMA_B | eDMA channel Interrupts 56-63                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |

Table continues on the next page...

**Table 4-4. Interrupt vectors and request sources (continued)**

| Offset | Vector | Source                                                                                                                                                                                                                                                                                                                    | Module     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|--------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x1ED0 | 493    | eTPU_GCR_CISR_ENG1[CIS24]                                                                                                                                                                                                                                                                                                 | eTPU_C     | eTPU Engine C Channel 24 Interrupt Status                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 0x1EE0 | 494    | eTPU_GCR_CISR_ENG1[CIS25]                                                                                                                                                                                                                                                                                                 | eTPU_C     | eTPU Engine C Channel 25 Interrupt Status                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 0x1EF0 | 495    | eTPU_GCR_CISR_ENG1[CIS26]                                                                                                                                                                                                                                                                                                 | eTPU_C     | eTPU Engine C Channel 26 Interrupt Status                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 0x1F00 | 496    | eTPU_GCR_CISR_ENG1[CIS27]                                                                                                                                                                                                                                                                                                 | eTPU_C     | eTPU Engine C Channel 27 Interrupt Status                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 0x1F10 | 497    | eTPU_GCR_CISR_ENG1[CIS28]                                                                                                                                                                                                                                                                                                 | eTPU_C     | eTPU Engine C Channel 28 Interrupt Status                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 0x1F20 | 498    | eTPU_GCR_CISR_ENG1[CIS29]                                                                                                                                                                                                                                                                                                 | eTPU_C     | eTPU Engine C Channel 29 Interrupt Status                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 0x1F30 | 499    | eTPU_GCR_CISR_ENG1[CIS30]                                                                                                                                                                                                                                                                                                 | eTPU_C     | eTPU Engine C Channel 30 Interrupt Status                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 0x1F40 | 500    | eTPU_GCR_CISR_ENG1[CIS31]                                                                                                                                                                                                                                                                                                 | eTPU_C     | eTPU Engine C Channel 31 Interrupt Status                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 0x1F50 | 501    | SWTB_IR[TIF]                                                                                                                                                                                                                                                                                                              | Watchdog B | Software Watchdog B Interrupt flag                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 0x1F60 | 502    | SEMA4_CP0INE[0:15]                                                                                                                                                                                                                                                                                                        | Semaphore  | Core 0 requested semaphore has unlocked                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 0x1F70 | 503    | SEMA4_CP1INE[0:15]                                                                                                                                                                                                                                                                                                        | Semaphore  | Core 1 requested semaphore has unlocked                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 0x1F80 | 504    | CSE_IR[CIF]                                                                                                                                                                                                                                                                                                               | CSE        | CSE Interrupt                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 0x1F90 | 505    | ESCIF_SR[TDRE]<br>ESCIF_SR[TC]<br>ESCIF_SR[RDRF]<br>ESCIF_SR[IDLE]<br>ESCIF_SR[OR]<br>ESCIF_SR[NF]<br>ESCIF_SR[FE]<br>ESCIF_SR[PF]<br>ESCIF_SR[BERR]<br>ESCIF_SR[RXRDY]<br>ESCIF_SR[TXRDY]<br>ESCIF_SR[LWAKE]<br>ESCIF_SR[STO]<br>ESCIF_SR[PBERR]<br>ESCIF_SR[CERR]<br>ESCIF_SR[CKERR]<br>ESCIF_SR[FRC]<br>ESCIF_SR[OVFL] | eSCI_F     | Combined Interrupt Requests of ESCI Module F:<br><br>Transmit Data Register Empty, Transmit Complete, Receive Data Register Full, Idle line, Overrun, Noise Flag, Framing Error Flag, and Parity Error Flag interrupt requests, SCI Status Register 2 Bit Error interrupt request, LIN Status Register 1 Receive Data Ready, Transmit Data Ready, Received LIN Wakeup Signal, Slave TimeOut, Physical Bus Error, CRC Error, Checksum Error, Frame Complete interrupts requests, and LIN Status Register 2 Receive Register Overflow |
| 0x1FA0 | 506    | DSPIE_SR[TFUF]<br>DSPIE_SR[RFOF]<br>DSPIE_SR[SPEF]<br>DSPIE_SR[DPEF]                                                                                                                                                                                                                                                      | DSPI_E     | DSPI_E combined overrun and parity error interrupt requests:<br><br>Transmit FIFO Underflow/Receive FIFO Overflow SPI Parity Error/DSI Parity Error                                                                                                                                                                                                                                                                                                                                                                                 |
| 0x1FB0 | 507    | DSPIE_SR[EOQF]                                                                                                                                                                                                                                                                                                            | DSPI_E     | DSPI_E transmit FIFO End of Queue Flag                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 0x1FC0 | 508    | DSPIE_SR[TFFF]                                                                                                                                                                                                                                                                                                            | DSPI_E     | DSPI_E Transmit FIFO Fill Flag                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 0x1FD0 | 509    | DSPIE_SR[TCF]   DSPIE_SR[DDIF]                                                                                                                                                                                                                                                                                            | DSPI_E     | DSPI_E Transfer Complete/DSI Data Match Flag                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |

Table continues on the next page...

**Table 4-4. Interrupt vectors and request sources (continued)**

| Offset | Vector | Source                | Module | Description                    |
|--------|--------|-----------------------|--------|--------------------------------|
| 0x1FE0 | 510    | DSPIE_SR[RDFD]        | DSPI_E | DSPI_E Receive FIFO Drain Flag |
| 0x1FF0 | 511    | STCU2_INT_FLG[MBIFLG] | STCU   | MBIST interrupt                |

## 4.3 DMA request mapping

This chip has two separate eDMA controllers: eDMA\_A and eDMA\_B. Each has 64 channels.

The following tables show the sources of DMA requests for each controller.

**Table 4-5. eDMA\_A channel request allocation**

| Channel | Requesting source                                             | Description                                                                                                                  |
|---------|---------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|
| 0       | EQADC_A_FISR0[CFFF0]                                          | EQADC_A Command FIFO 0 Fill Flag                                                                                             |
| 1       | EQADC_A_FISR0[RDFD0]                                          | EQADC_A Receive FIFO 0 Drain Flag                                                                                            |
| 2       | EQADC_A_FISR1[CFFF1]                                          | EQADC_A Command FIFO 1 Fill Flag                                                                                             |
| 3       | EQADC_A_FISR1[RDFD1]                                          | EQADC_A Receive FIFO 1 Drain Flag                                                                                            |
| 4       | EQADC_A_FISR2[CFFF2]                                          | EQADC_A Command FIFO 2 Fill Flag                                                                                             |
| 5       | EQADC_A_FISR2[RDFD2]                                          | EQADC_A Receive FIFO 2 Drain Flag                                                                                            |
| 6       | EQADC_A_FISR3[CFFF3]                                          | EQADC_A Command FIFO 3 Fill Flag                                                                                             |
| 7       | EQADC_A_FISR3[RDFD3]                                          | EQADC_A Receive FIFO 3 Drain Flag                                                                                            |
| 8       | EQADC_A_FISR4[CFFF4]                                          | EQADC_A Command FIFO 4 Fill Flag                                                                                             |
| 9       | EQADC_A_FISR4[RDFD4]                                          | EQADC_A Receive FIFO 4 Drain Flag                                                                                            |
| 10      | EQADC_A_FISR5[CFFF5]                                          | EQADC_A Command FIFO 5 Fill Flag                                                                                             |
| 11      | EQADC_A_FISR5[RDFD5]                                          | EQADC_A Receive FIFO 5 Drain Flag                                                                                            |
| 12      | DSPI_B_SR[TFFF]                                               | DSPI_B Transmit FIFO Fill Flag                                                                                               |
| 13      | DSPI_B_SR[RDFD]                                               | DSPI_B Receive FIFO Drain Flag                                                                                               |
| 14      | DSPI_C_SR[TFFF]                                               | DSPI_C Transmit FIFO Fill Flag                                                                                               |
| 15      | DSPI_C_SR[RDFD]                                               | DSPI_C Receive FIFO Drain Flag                                                                                               |
| 16      | DSPI_D_SR[TFFF]                                               | DSPI_D Transmit FIFO Fill Flag                                                                                               |
| 17      | DSPI_D_SR[RDFD]                                               | DSPI_D Receive FIFO Drain Flag                                                                                               |
| 18      | eSCI_A_IFSR1[TDRE]    eSCI_A_IFSR1[TC]    eSCI_A_IFSR2[TXRDY] | eSCI_A combined DMA request of the Transmit Data Register Empty, Transmit Complete, and LIN Transmit Data Ready DMA requests |
| 19      | eSCI_A_IFSR1[RDRF]    eSCI_A_IFSR2[RXRDY]                     | eSCI_A combined DMA request of the Receive Data Register Full and LIN Receive Data Ready DMA requests                        |
| 20      | eMIOS_0_GFLAG[F0]                                             | eMIOS_0 channel 0 Flag                                                                                                       |
| 21      | eMIOS_0_GFLAG[F1]                                             | eMIOS_0 channel 1 Flag                                                                                                       |
| 22      | eMIOS_0_GFLAG[F2]                                             | eMIOS_0 channel 2 Flag                                                                                                       |

Table continues on the next page...

**Table 4-5. eDMA\_A channel request allocation (continued)**

| Channel | Requesting source                                             | Description                                                                                                                  |
|---------|---------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|
| 23      | eMIOS_0_GFLAG[F3]                                             | eMIOS_0 channel 3 Flag                                                                                                       |
| 24      | eMIOS_0_GFLAG[F4]                                             | eMIOS_0 channel 4 Flag                                                                                                       |
| 25      | eMIOS_1_GFLAG[F0]                                             | eMIOS_1 channel 0 Flag                                                                                                       |
| 26      | eMIOS_1_GFLAG[F1]                                             | eMIOS_1 channel 1 Flag                                                                                                       |
| 27      | eTPU_AB_GCR_CDTRSR_ENG1[DTRS0]                                | eTPU_A channel 0 Data Transfer Request Status                                                                                |
| 28      | eTPU_AB_GCR_CDTRSR_ENG1[DTRS1]                                | eTPU_A channel 1 Data Transfer Request Status                                                                                |
| 29      | eTPU_AB_GCR_CDTRSR_ENG1[DTRS2]                                | eTPU_A channel 2 Data Transfer Request Status                                                                                |
| 30      | eTPU_AB_GCR_CDTRSR_ENG1[DTRS14]                               | eTPU_A channel 14 Data Transfer Request Status                                                                               |
| 31      | eTPU_AB_GCR_CDTRSR_ENG1[DTRS15]                               | eTPU_A channel 15 Data Transfer Request Status                                                                               |
| 32      | DSPI_A_SR[TFFF]                                               | DSPI_A Transmit FIFO Fill Flag                                                                                               |
| 33      | DSPI_A_SR[RDFD]                                               | DSPI_A Receive FIFO Drain Flag                                                                                               |
| 34      | eSCI_B_IFSR1[TDRE]    eSCI_B_IFSR1[TC]    eSCI_B_IFSR2[TXRDY] | eSCI_B combined DMA request of the Transmit Data Register Empty, Transmit Complete, and LIN Transmit Data Ready DMA requests |
| 35      | eSCI_B_IFSR1[RDRF]    eSCI_B_IFSR2[RXRDY]                     | eSCI_B combined DMA request of the Receive Data Register Full and LIN Receive Data Ready DMA requests                        |
| 36      | eMIOS_0_GFLAG[F6]                                             | eMIOS_0 channel 6 Flag                                                                                                       |
| 37      | eMIOS_0_GFLAG[F7]                                             | eMIOS_0 channel 7 Flag                                                                                                       |
| 38      | eMIOS_1_GFLAG[F2]                                             | eMIOS_1 channel 2 Flag                                                                                                       |
| 39      | eMIOS_1_GFLAG[F3]                                             | eMIOS_1 channel 3 Flag                                                                                                       |
| 40      | eMIOS_0_GFLAG[F16]                                            | eMIOS_0 channel 16 Flag                                                                                                      |
| 41      | eMIOS_0_GFLAG[F17]                                            | eMIOS_0 channel 17 Flag                                                                                                      |
| 42      | eMIOS_0_GFLAG[F18]                                            | eMIOS_0 channel 18 Flag                                                                                                      |
| 43      | eMIOS_0_GFLAG[F19]                                            | eMIOS_0 channel 19 Flag                                                                                                      |
| 44      | eTPU_AB_GCR_CDTRSR_ENG1[DTRS12]                               | eTPU_A channel 12 Data Transfer Request Status                                                                               |
| 45      | eTPU_AB_GCR_CDTRSR_ENG1[DTRS13]                               | eTPU_A channel 13 Data Transfer Request Status                                                                               |
| 46      | eTPU_AB_GCR_CDTRSR_ENG1[DTRS28]                               | eTPU_A channel 28 Data Transfer Request Status                                                                               |
| 47      | eTPU_AB_GCR_CDTRSR_ENG1[DTRS29]                               | eTPU_A channel 29 Data Transfer Request Status                                                                               |
| 48      | SIU_EISR[EIF0]                                                | SIU External Interrupt Flag 0                                                                                                |
| 49      | SIU_EISR[EIF1]                                                | SIU External Interrupt Flag 1                                                                                                |
| 50      | SIU_EISR[EIF2]                                                | SIU External Interrupt Flag 2                                                                                                |
| 51      | SIU_EISR[EIF3]                                                | SIU External Interrupt Flag 3                                                                                                |
| 52      | eTPU_AB_GCR_CDTRSR_ENG2[DTRS0]                                | eTPU_B channel 0 Data Transfer Request Status                                                                                |
| 53      | eTPU_AB_GCR_CDTRSR_ENG2[DTRS1]                                | eTPU_B channel 1 Data Transfer Request Status                                                                                |
| 54      | eTPU_AB_GCR_CDTRSR_ENG2[DTRS2]                                | eTPU_B channel 2 Data Transfer Request Status                                                                                |
| 55      | eTPU_AB_GCR_CDTRSR_ENG2[DTRS3]                                | eTPU_B channel 3 Data Transfer Request Status                                                                                |
| 56      | eTPU_AB_GCR_CDTRSR_ENG2[DTRS12]                               | eTPU_B channel 12 Data Transfer Request Status                                                                               |
| 57      | eTPU_AB_GCR_CDTRSR_ENG2[DTRS13]                               | eTPU_B channel 13 Data Transfer Request Status                                                                               |
| 58      | eTPU_AB_GCR_CDTRSR_ENG2[DTRS14]                               | eTPU_B channel 14 Data Transfer Request Status                                                                               |
| 59      | eTPU_AB_GCR_CDTRSR_ENG2[DTRS15]                               | eTPU_B channel 15 Data Transfer Request Status                                                                               |

*Table continues on the next page...*

**Table 4-5. eDMA\_A channel request allocation (continued)**

| Channel | Requesting source               | Description                                    |
|---------|---------------------------------|------------------------------------------------|
| 60      | eTPU_AB_GCR_CDTRSR_ENG2[DTRS28] | eTPU_B channel 28 Data Transfer Request Status |
| 61      | eTPU_AB_GCR_CDTRSR_ENG2[DTRS29] | eTPU_B channel 29 Data Transfer Request Status |
| 62      | eTPU_AB_GCR_CDTRSR_ENG2[DTRS30] | eTPU_B channel 30 Data Transfer Request Status |
| 63      | eTPU_AB_GCR_CDTRSR_ENG2[DTRS31] | eTPU_B channel 31 Data Transfer Request Status |

**Table 4-6. eDMA\_B channel request allocation**

| Channel | Requesting source               | Description                                      |
|---------|---------------------------------|--------------------------------------------------|
| 0       | EQADC_B_FISR0[CFFF0]            | EQADC_B Command FIFO 0 Fill Flag                 |
| 1       | EQADC_B_FISR0[RFDF0]            | EQADC_B Result FIFO 0 Drain Flag                 |
| 2       | EQADC_B_FISR1[CFFF1]            | EQADC_B Command FIFO 1 Fill Flag                 |
| 3       | EQADC_B_FISR1[RFDF1]            | EQADC_B Result FIFO 1 Drain Flag                 |
| 4       | EQADC_B_FISR2[CFFF2]            | EQADC_B Command FIFO 2 Fill Flag                 |
| 5       | EQADC_B_FISR2[RFDF2]            | EQADC_B Result FIFO 2 Drain Flag                 |
| 6       | EQADC_B_FISR3[CFFF3]            | EQADC_B Command FIFO 3 Fill Flag                 |
| 7       | EQADC_B_FISR3[RFDF3]            | EQADC_B Result FIFO 3 Drain Flag                 |
| 8       | EQADC_B_FISR4[CFFF4]            | EQADC_B Command FIFO 4 Fill Flag                 |
| 9       | EQADC_B_FISR4[RFDF4]            | EQADC_B Result FIFO 4 Drain Flag                 |
| 10      | EQADC_B_FISR5[CFFF5]            | EQADC_B Command FIFO 5 Fill Flag                 |
| 11      | EQADC_B_FISR5[RFDF5]            | EQADC_B Result FIFO 5 Drain Flag                 |
| 36      | SDADC_0                         | SDADC_0 result ready                             |
| 37      | SDADC_1                         | SDADC_1 result ready                             |
| 38      | SDADC_2                         | SDADC_2 result ready                             |
| 39      | SDADC_3                         | SDADC_3 result ready                             |
| 40      | SRX_0 Fast Message Ready        | SENT Receiver Module 0 Fast Message Ready        |
| 41      | SRX_1 Fast Message Ready        | SENT Receiver Module 1 Fast Message Ready        |
| 42      | SRX_0 Slow Serial Message Ready | SENT Receiver Module 0 Slow Serial Message Ready |
| 43      | SRX_1 Slow Serial Message Ready | SENT Receiver Module 1 Slow Serial Message Ready |
| 44      | eTPU_C_GCR_CDTRSR_ENG1[DTRS0]   | eTPU_C Channel 0 Data Transfer Request Status    |
| 45      | eTPU_C_GCR_CDTRSR_ENG1[DTRS1]   | eTPU_C Channel 1 Data Transfer Request Status    |
| 46      | eTPU_C_GCR_CDTRSR_ENG1[DTRS2]   | eTPU_C Channel 2 Data Transfer Request Status    |
| 47      | eTPU_C_GCR_CDTRSR_ENG1[DTRS3]   | eTPU_C Channel 3 Data Transfer Request Status    |
| 48      | eTPU_C_GCR_CDTRSR_ENG1[DTRS4]   | eTPU_C Channel 4 Data Transfer Request Status    |
| 49      | eTPU_C_GCR_CDTRSR_ENG1[DTRS5]   | eTPU_C Channel 5 Data Transfer Request Status    |
| 50      | eTPU_C_GCR_CDTRSR_ENG1[DTRS6]   | eTPU_C Channel 6 Data Transfer Request Status    |
| 51      | eTPU_C_GCR_CDTRSR_ENG1[DTRS7]   | eTPU_C Channel 7 Data Transfer Request Status    |
| 60      | M_CAN_0                         | M_CAN_0 DMA Request                              |
| 61      | DSPI_E_SR[TFFF]                 | DSPI_E Transmit FIFO Fill Flag                   |
| 62      | DSPI_E_SR[RFDF]                 | DSPI_E Receive FIFO Drain Flag                   |
| 63      | M_CAN_1                         | M_CAN_1 DMA Request                              |







# Chapter 5

## Device Configuration Format (DCF) Records

### 5.1 Overview

Device Configuration Format (DCF) records are used by the System Status and Configuration Module (SSCM) to configure certain registers in the device during system boot while the reset signal is asserted.

There are two kinds of DCF records:

- Factory-written DCF records: written to UTEST flash memory by the factory.
- User-supplied DCF records: written to UTEST flash memory by the user.

The following figure is a high-level conceptual representation of the functionality of DCF records.

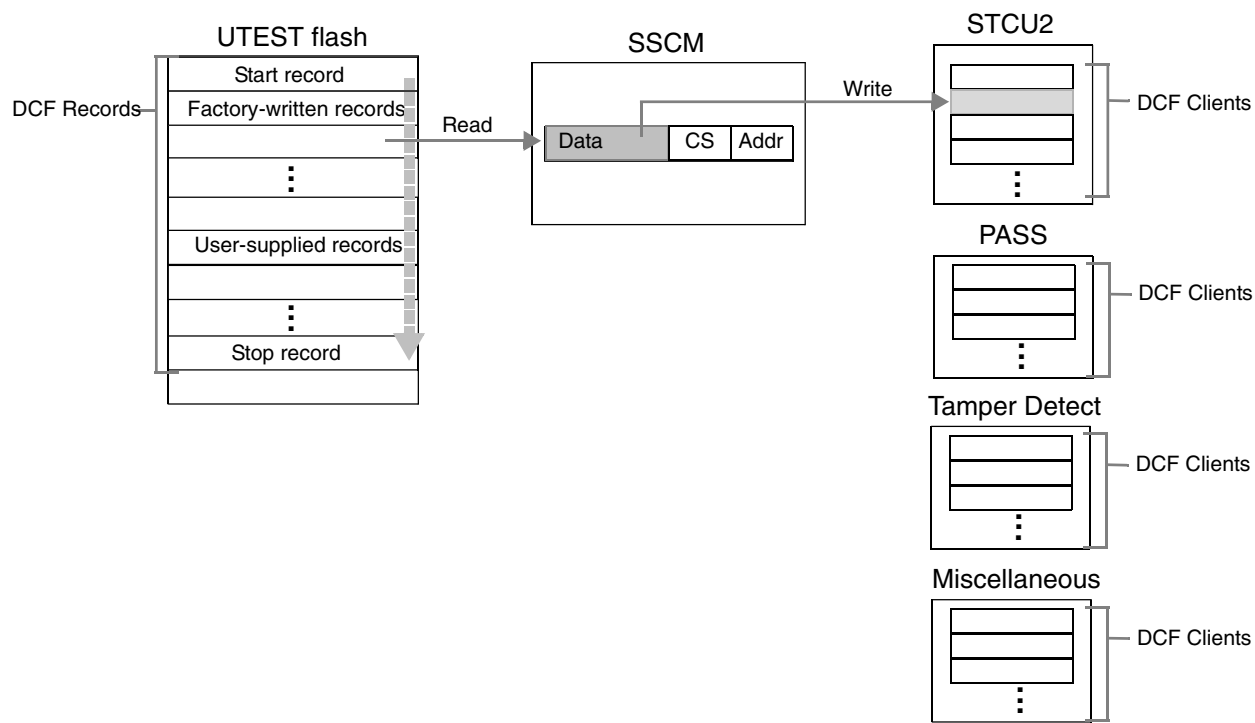


**Figure 5-1. Conceptual representation of DCF record functionality**

## 5.2 DCF records in boot sequence

The following is a high-level summary of the boot sequence leading to the use of DCF records:

1. When power is applied to a properly programmed device, the Power Management Controller (PMC) takes control of the device.
2. After the system power supplies have reached predefined levels, the flash memory array is initialized and the SSCM is enabled.
3. The SSCM reads the DCF records.
4. The SSCM writes the configuration information to the specified registers.
5. After registers are initialized: If the STCU2 is enabled by DCF record information, it begins executing the tasks enabled by its DCF record configuration.

## 5.3 DCF records location

Factory-written DCF records start at the first address allocated to the DCF records in the UTEST flash memory area. See the UTEST flash memory map.

User-supplied UTEST DCF records may be added at the next location in the UTEST memory map immediately following the factory-written UTEST DCF records.

## 5.4 DCF record structure

A DCF record is a contiguous double-word (64-bit) entry consisting of the following:

- Control word: Information to locate the corresponding DCF client internal to the device (pointer to the location of a register internal to the device)
- Data word: Data to be written to that client

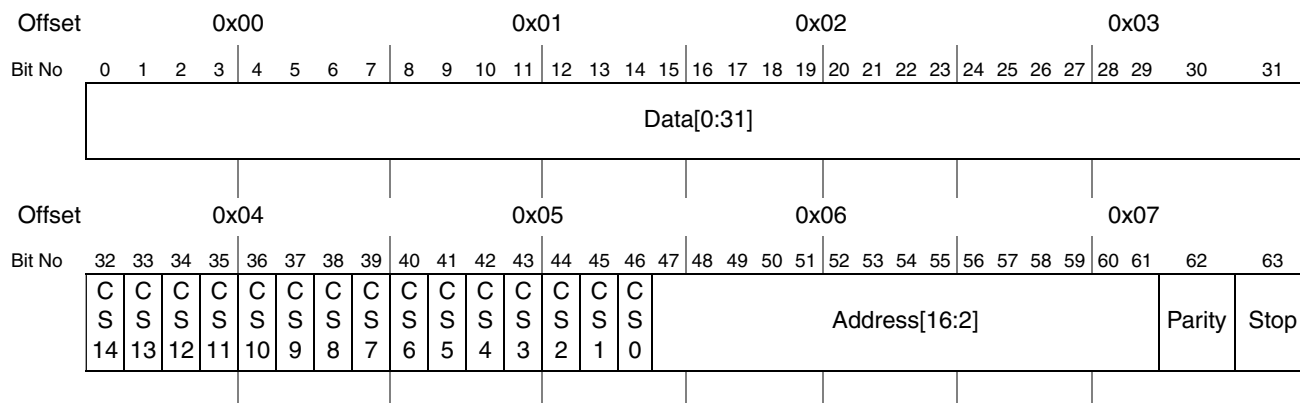


Figure 5-2. DCF record structure

Table 5-1. DCF record field descriptions

| Field | Name          | Description                                                                                                                                                                                                                                                                                                                                                                                                  |
|-------|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0-31  | Data[0:31]    | 32 bits of data to be written to the DCF client                                                                                                                                                                                                                                                                                                                                                              |
| 32-46 | CS $n$        | Chip Select $n$<br><br>For each DCF record, one Chip Select bit is asserted (1b) to select the target module for the DCF client. All other Chip Select bits per DCF record should be negated (0b).                                                                                                                                                                                                           |
| 47-61 | Address[16:2] | Address of the DCF client within the selected module<br><br><b>NOTE:</b> Address decoding for DCF clients may not match the standard software address map decoding.<br><br>Details of DCF client addresses are defined in each applicable module chapter.                                                                                                                                                    |
| 62    | Parity        | Parity bit for the DCF record                                                                                                                                                                                                                                                                                                                                                                                |
| 63    | Stop          | Stop bit<br><br>Indicates the end of the list of DCF records.<br><br>0 indicates the DCF record is not the end of the list<br><br>1 indicates the DCF record is the end of the list<br><br><b>NOTE:</b> The erased state of flash memory is 0xFFFF_FFFF_FFFF_FFFF. As a result, the list ends with the first unprogrammed double word. This location can be programmed with a new record to extend the list. |

## 5.5 DCF records sequence

DCF records must appear as contiguous series of entries programmed from the beginning of the UTEST flash memory area in the following order.

1. **Start DCF record:** The first DCF record must be a start record. This record must be found at the beginning of a DCF area in flash memory to indicate to the device that the following records must be processed.
2. **DCF records containing configuration data:** DCF records containing configuration data must immediately follow the start record with no blank records in between.
3. **Stop DCF record:** The end of the configuration records is indicated by the stop record. A record with the stop bit set is a stop record. All other bits are ignored. No DCF records following the stop record are processed.

SSCM recognizes the following record as a start record.

**Table 5-2. DCF start record**

|             |              |
|-------------|--------------|
| 0x00 (0:31) | 0x04 (32:63) |
| 0x05AA_55AF | 0x0000_0000  |

The start bit is set by the factory at the beginning of the UTEST flash memory area.

SSCM recognizes the following record as a stop record.

**Table 5-3. DCF stop record**

|         |         |    |
|---------|---------|----|
| 0:31    | 32:62   | 63 |
| Ignored | Ignored | 1  |

The stop bit is never set by the factory, since this would not allow the user to add DCF records. The flash memory location following the last factory-written UTEST DCF record is unprogrammed, with the content of 0xFFFF\_FFFF. This is interpreted as the stop record, since the stop bit is set.

User-supplied DCF records may be added in a contiguous manner immediately following the factory-written DCF records. There must never be an unprogrammed record in the series of DCF records as it will be interpreted as a stop record.

The following table shows the series of DCF records when  $n$  data records are stored in the UTEST flash.

**Table 5-4. Series of DCF records in UTEST flash memory**

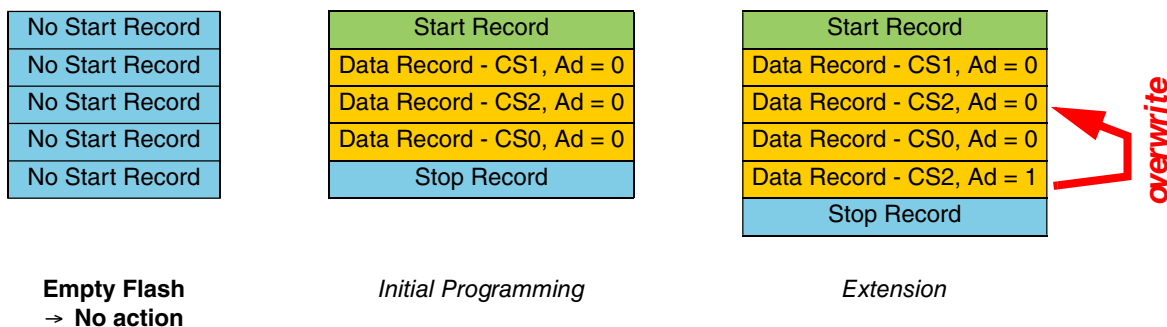
| Record type  | ADDR offset | DATA        |            |      |        |
|--------------|-------------|-------------|------------|------|--------|
| Start record | 0x00        | 0x05AA_55AF |            |      |        |
|              | 0x04        | 0x0000_0000 |            |      | STOP=0 |
| Data records | 0x08        | WDATA[31:0] |            |      |        |
|              | 0x0C        | CS[14:0]    | ADDR[16:2] | PRTY | STOP=0 |

*Table continues on the next page...*

**Table 5-4. Series of DCF records in UTEST flash memory (continued)**

| Record type | ADDR offset  | DATA        |            |      |        |
|-------------|--------------|-------------|------------|------|--------|
|             | 0x10         | WDATA[31:0] |            |      |        |
|             | 0x14         | CS[14:0]    | ADDR[16:2] | PRTY | STOP=0 |
|             | ...          | ...         |            |      |        |
| Stop record | $8n-1 + 0x0$ | Reserved    |            |      |        |
|             | $8n-1 + 0x4$ | Reserved    |            |      | 1      |
|             | $8n + 0x0$   |             |            |      |        |
|             | $8n + 0x4$   |             |            |      |        |

It is possible for more than one DCF Record to write to the same DCF client. In this case the later record usually overrides a DCF client value set by a previous record. However, not all DCF clients allow overwrites; this depends on the DCF client implementation.

**Figure 5-3. Appending DCF records**

## 5.6 DCF client Chip Select module assignments

The following table shows the Chip Select (CS) for each group of DCF clients. A CS can be assigned to one module or multiple modules, as indicated. CSs not listed are reserved.

**Table 5-5. DCF client CS module assignments**

| CS    | DCF client target module(s) |
|-------|-----------------------------|
| CS[0] | Reserved                    |
| CS[1] | SSCM (Reserved)             |
| CS[2] | STCU2                       |
| CS[3] | PASS                        |
| CS[4] | Tamper Detect (TDM)         |

Table continues on the next page...

**Table 5-5. DCF client CS module assignments (continued)**

| CS    | DCF client target module(s)                                                                                                                                   |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CS[5] | Reserved                                                                                                                                                      |
| CS[6] | Reserved                                                                                                                                                      |
| CS[7] | Miscellaneous<br>CSE Secure Boot Configurations<br>Analog Oscillator Configurations<br>ERROR Functions Pads Configuration<br>Reset Escalation<br>PMC Trimming |

## 5.7 DCF clients

DCF clients are 32-bit-wide hardware registers inside a module that receive and store the data from a DCF record. This stored data is used to initialize registers and to configure features. Their characteristics include:

- DCF clients have a default value before any DCF records are written.
- DCF clients are not required to implement all 32 bits.

The following table below shows the DCF clients that are available in this chip's UTEST area of flash memory. The following list of terms defines the meaning of the table's column headings.

### DCF CS: Chip Select

The CS field is a 15-bit field with each bit in the field assigned to a particular module. These bits act as module select signals for the internal modules. Only one bit in this field should be set in a DCF record.

### DCF Address[10:2]

The Address[16:2] field of a DCF record specifies an internal address for a DCF client. The following table provides the Address[10:2] portion of the field. The address is used only by the SSCM when interpreting DCF records and writing the data to specified modules.

### DCF client description

This entry contains the name of the DCF client.

### DCF client reset value

This field contains one of the following:

- the reset value of the DCF client before the SSCM writes the client with the associated DCF record
- a factory-programmed value

#### DCF client target module

This entry identifies the module that uses the DCF client information.

#### DCF client special strategy

Special requirements apply to how some DCF clients are written. The SSCM examines the DCF client to determine any correct special strategy to use and then appropriately writes a DCF record to that client.

##### None

No special DCF strategy is used.

##### Parity

On this chip, parity is implemented only for PMC DCF clients.

##### Triple Voted

DCF clients that use the Triple Voted strategy have three copies of the register. The SSCM writes to all three registers in a single write cycle. The outputs of the three registers are majority voted together to determine the correct data value. Triple voting allows a bit-flip error to occur without changing the DCF client output data.

##### write\_once

A register using the Write Once strategy can only be written once. The DCF client ignores subsequent writes.

### CAUTION

Plan carefully before writing write-once DCF clients. Any DCF client that has a write-once attribute cannot be reprogrammed by user software until the next reset or by more than one DCF record. Subsequent DCF records are ignored.

##### wr0\_only: Write 0 only

A bit in a DCF client can be written only from a logic 1 to a logic 0. An attempt to write a bit with this attribute to a logic 1 is ignored.

##### wr1\_only: Write 1 only

A bit in a DCF client can be written only from a logic 0 to a logic 1. An attempt to write a bit with this attribute to a logic 0 is ignored.

Any UTEST DCF client can be programmed as UTEST non-differential.

**Table 5-6. UTEST DCF clients**

| DCF CS[14:0]                                       | DCF Address[10:2] | DCF client description | DCF client reset value (hex) | DCF client target module | DCF client special strategy |
|----------------------------------------------------|-------------------|------------------------|------------------------------|--------------------------|-----------------------------|
| STCU2: support for 6 LBIST partitions; 64-bit MISR |                   |                        |                              |                          |                             |
| 000_0000_0000_0100                                 | 0_0000_0010       | STCU_SKC               | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100                                 | 0_0000_0000       | STCU_RUN               | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100                                 | 0_0000_0011       | STCU_CFG               | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100                                 | 0_0000_0100       | STCU_PLL_CFG           | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100                                 | 0_0000_0101       | STCU_WDG               | 0000_FFFF                    | STCU                     | None                        |
| 000_0000_0000_0100                                 | 0_0000_0111       | Reserved               |                              |                          |                             |
| 000_0000_0000_0100                                 | 0_0000_1010       | STCU_ERR_FM            | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100                                 | 0_0000_1111       | STCU_LBRMSW            | 0000_003F                    | STCU                     | None                        |
| 000_0000_0000_0100                                 | 0_0001_0000       | STCU_LBUFM             | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100                                 | 0_0001_1101       | STCU_MBUFML            | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100                                 | 0_0001_1110       | STCU_MBUFMM            | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100                                 | 0_0001_1111       | STCU_MBUFMH            | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100                                 | 0_0100_0000       | STCU_LB_CTRL_0         | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100                                 | 0_0100_0001       | STCU_LB_PCS_0          | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100                                 | 0_0100_0010       | Reserved               |                              |                          |                             |
| 000_0000_0000_0100                                 | 0_0100_0011       | Reserved               |                              |                          |                             |
| 000_0000_0000_0100                                 | 0_0100_0100       | STCU_LB_MISREL_0       | FFFF_FFFF                    | STCU                     | None                        |
| 000_0000_0000_0100                                 | 0_0100_0101       | STCU_LB_MISREH_0       | FFFF_FFFF                    | STCU                     | None                        |
| 000_0000_0000_0100                                 | 0_0100_1000       | STCU_LB_MISRELSW_0     | FFFF_FFFF                    | STCU                     | None                        |
| 000_0000_0000_0100                                 | 0_0100_1001       | STCU_LB_MISREHSW_0     | FFFF_FFFF                    | STCU                     | None                        |
| 000_0000_0000_0100                                 | 0_0101_0000       | STCU_LB_CTRL_1         | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100                                 | 0_0101_0001       | STCU_LB_PCS_1          | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100                                 | 0_0101_0010       | Reserved               |                              |                          |                             |
| 000_0000_0000_0100                                 | 0_0101_0011       | Reserved               |                              |                          |                             |
| 000_0000_0000_0100                                 | 0_0101_0100       | STCU_LB_MISREL_1       | FFFF_FFFF                    | STCU                     | None                        |
| 000_0000_0000_0100                                 | 0_0101_0101       | STCU_LB_MISREH_1       | FFFF_FFFF                    | STCU                     | None                        |
| 000_0000_0000_0100                                 | 0_0101_1000       | STCU_LB_MISRELSW_1     | FFFF_FFFF                    | STCU                     | None                        |
| 000_0000_0000_0100                                 | 0_0101_1001       | STCU_LB_MISREHSW_1     | FFFF_FFFF                    | STCU                     | None                        |
| 000_0000_0000_0100                                 | 0_0110_0000       | STCU_LB_CTRL_2         | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100                                 | 0_0110_0001       | STCU_LB_PCS_2          | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100                                 | 0_0110_0010       | Reserved               |                              |                          |                             |
| 000_0000_0000_0100                                 | 0_0110_0011       | Reserved               |                              |                          |                             |
| 000_0000_0000_0100                                 | 0_0110_0100       | STCU_LB_MISREL_2       | FFFF_FFFF                    | STCU                     | None                        |

Table continues on the next page...



Table 5-6. UTEST DCF clients (continued)

| DCF CS[14:0]       | DCF Address[10:2] | DCF client description | DCF client reset value (hex) | DCF client target module | DCF client special strategy |
|--------------------|-------------------|------------------------|------------------------------|--------------------------|-----------------------------|
| 000_0000_0000_0100 | 0_0110_0101       | STCU_LB_MISREH_2       | FFFF_FFFF                    | STCU                     | None                        |
| 000_0000_0000_0100 | 0_0110_1000       | STCU_LB_MISRELSW_2     | FFFF_FFFF                    | STCU                     | None                        |
| 000_0000_0000_0100 | 0_0110_1001       | STCU_LB_MISREHSW_2     | FFFF_FFFF                    | STCU                     | None                        |
| 000_0000_0000_0100 | 0_0111_0000       | STCU_LB_CTRL_3         | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 0_0111_0001       | STCU_LB_PCS_3          | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 0_0111_0010       | Reserved               |                              |                          |                             |
| 000_0000_0000_0100 | 0_0111_0011       | Reserved               |                              |                          |                             |
| 000_0000_0000_0100 | 0_0111_0100       | STCU_LB_MISREL_3       | FFFF_FFFF                    | STCU                     | None                        |
| 000_0000_0000_0100 | 0_0111_0101       | STCU_LB_MISREH_3       | FFFF_FFFF                    | STCU                     | None                        |
| 000_0000_0000_0100 | 0_0111_1000       | STCU_LB_MISRELSW_3     | FFFF_FFFF                    | STCU                     | None                        |
| 000_0000_0000_0100 | 0_0111_1001       | STCU_LB_MISREHSW_3     | FFFF_FFFF                    | STCU                     | None                        |
| 000_0000_0000_0100 | 0_1000_0000       | STCU_LB_CTRL_4         | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 0_1000_0001       | STCU_LB_PCS_4          | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 0_1000_0010       | Reserved               |                              |                          |                             |
| 000_0000_0000_0100 | 0_1000_0011       | Reserved               |                              |                          |                             |
| 000_0000_0000_0100 | 0_1000_0100       | STCU_LB_MISREL_4       | FFFF_FFFF                    | STCU                     | None                        |
| 000_0000_0000_0100 | 0_1000_0101       | STCU_LB_MISREH_4       | FFFF_FFFF                    | STCU                     | None                        |
| 000_0000_0000_0100 | 0_1000_1000       | STCU_LB_MISRELSW_4     | FFFF_FFFF                    | STCU                     | None                        |
| 000_0000_0000_0100 | 0_1000_1001       | STCU_LB_MISREHSW_4     | FFFF_FFFF                    | STCU                     | None                        |
| 000_0000_0000_0100 | 0_1001_0000       | STCU_LB_CTRL_5         | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 0_1001_0001       | STCU_LB_PCS_5          | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 0_1001_0010       | Reserved               |                              |                          |                             |
| 000_0000_0000_0100 | 0_1001_0011       | Reserved               |                              |                          |                             |
| 000_0000_0000_0100 | 0_1001_0100       | STCU_LB_MISREL_5       | FFFF_FFFF                    | STCU                     | None                        |
| 000_0000_0000_0100 | 0_1001_0101       | STCU_LB_MISREH_5       | FFFF_FFFF                    | STCU                     | None                        |
| 000_0000_0000_0100 | 0_1001_1000       | STCU_LB_MISRELSW_5     | FFFF_FFFF                    | STCU                     | None                        |
| 000_0000_0000_0100 | 0_1001_1001       | STCU_LB_MISREHSW_5     | FFFF_FFFF                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1000_0000       | STCU_MB_CTRL_0         | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1000_0001       | STCU_MB_CTRL_1         | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1000_0010       | STCU_MB_CTRL_2         | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1000_0011       | STCU_MB_CTRL_3         | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1000_0100       | STCU_MB_CTRL_4         | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1000_0101       | STCU_MB_CTRL_5         | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1000_0110       | STCU_MB_CTRL_6         | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1000_0111       | STCU_MB_CTRL_7         | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1000_1000       | STCU_MB_CTRL_8         | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1000_1001       | STCU_MB_CTRL_9         | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1000_1010       | STCU_MB_CTRL_10        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1000_1011       | STCU_MB_CTRL_11        | 0000_0000                    | STCU                     | None                        |

Table continues on the next page...

Table 5-6. UTEST DCF clients (continued)

| DCF CS[14:0]       | DCF Address[10:2] | DCF client description | DCF client reset value (hex) | DCF client target module | DCF client special strategy |
|--------------------|-------------------|------------------------|------------------------------|--------------------------|-----------------------------|
| 000_0000_0000_0100 | 1_1000_1100       | STCU_MB_CTRL_12        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1000_1101       | STCU_MB_CTRL_13        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1000_1110       | STCU_MB_CTRL_14        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1000_1111       | STCU_MB_CTRL_15        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1001_0000       | STCU_MB_CTRL_16        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1001_0001       | STCU_MB_CTRL_17        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1001_0010       | STCU_MB_CTRL_18        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1001_0011       | STCU_MB_CTRL_19        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1001_0100       | STCU_MB_CTRL_20        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1001_0101       | STCU_MB_CTRL_21        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1001_0110       | STCU_MB_CTRL_22        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1001_0111       | STCU_MB_CTRL_23        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1001_1000       | STCU_MB_CTRL_24        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1001_1001       | STCU_MB_CTRL_25        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1001_1010       | STCU_MB_CTRL_26        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1001_1011       | STCU_MB_CTRL_27        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1001_1100       | STCU_MB_CTRL_28        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1001_1101       | STCU_MB_CTRL_29        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1001_1110       | STCU_MB_CTRL_30        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1001_1111       | STCU_MB_CTRL_31        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1010_0000       | STCU_MB_CTRL_32        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1010_0001       | STCU_MB_CTRL_33        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1010_0010       | STCU_MB_CTRL_34        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1010_0011       | STCU_MB_CTRL_35        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1010_0100       | STCU_MB_CTRL_36        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1010_0101       | STCU_MB_CTRL_37        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1010_0110       | STCU_MB_CTRL_38        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1010_0111       | STCU_MB_CTRL_39        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1010_1000       | STCU_MB_CTRL_40        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1010_1001       | STCU_MB_CTRL_41        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1010_1010       | STCU_MB_CTRL_42        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1010_1011       | STCU_MB_CTRL_43        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1010_1100       | STCU_MB_CTRL_44        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1010_1101       | STCU_MB_CTRL_45        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1010_1110       | STCU_MB_CTRL_46        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1010_1111       | STCU_MB_CTRL_47        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1011_0000       | STCU_MB_CTRL_48        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1011_0001       | STCU_MB_CTRL_49        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100 | 1_1011_0010       | STCU_MB_CTRL_50        | 0000_0000                    | STCU                     | None                        |

Table continues on the next page...

Table 5-6. UTEST DCF clients (continued)

| DCF CS[14:0]        | DCF Address[10:2]       | DCF client description | DCF client reset value (hex) | DCF client target module | DCF client special strategy |
|---------------------|-------------------------|------------------------|------------------------------|--------------------------|-----------------------------|
| 000_0000_0000_0100  | 1_1011_0011             | STCU_MB_CTRL_51        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100  | 1_1011_0100             | STCU_MB_CTRL_52        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100  | 1_1011_0101             | STCU_MB_CTRL_53        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100  | 1_1011_0110             | STCU_MB_CTRL_54        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100  | 1_1011_0111             | STCU_MB_CTRL_55        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100  | 1_1011_1000             | STCU_MB_CTRL_56        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100  | 1_1011_1001             | STCU_MB_CTRL_57        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100  | 1_1011_1010             | STCU_MB_CTRL_58        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100  | 1_1011_1011             | STCU_MB_CTRL_59        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100  | 1_1011_1100             | STCU_MB_CTRL_60        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100  | 1_1011_1101             | STCU_MB_CTRL_61        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100  | 1_1011_1110             | STCU_MB_CTRL_62        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100  | 1_1011_1111             | STCU_MB_CTRL_63        | 0000_0000                    | STCU                     | None                        |
| 000_0000_0000_0100  | 1_1100_0000             | STCU_MB_CTRL_64        | 0000_0000                    | STCU                     | None                        |
| PASS                |                         |                        |                              |                          |                             |
| 000_0000_0000_1000  | 0_0000_0001-0_0010_1011 | Reserved               |                              |                          |                             |
| 000_0000_0000_1000  | 0_0010_1100             | Censorship             | 0000_0000                    | PASS                     | None                        |
| 000_0000_0000_1000  | 0_0010_1101-0_0010_1111 | Reserved               |                              |                          |                             |
| 000_0000_0000_1000  | 0_0011_0000             | Production Disable     | 0000_0000                    | PASS                     | write_once                  |
| 000_0000_0000_1000  | 0_0100_0000             | LOCK0_PG0              | FFFF_FFFF                    | PASS                     | None                        |
| 000_0000_0000_1000  | 0_0100_0001             | LOCK1_PG0              | FFFF_FFFF                    | PASS                     | None                        |
| 000_0000_0000_1000  | 0_0100_0010             | LOCK2_PG0              | FFFF_FFFF                    | PASS                     | None                        |
| 000_0000_0000_1000  | 0_0100_0011             | LOCK3_PG0              | FFFF_FFFF                    | PASS                     | None                        |
| 000_0000_0000_1000  | 0_0100_0100             | LOCK0_PG1              | FFFF_FFFF                    | PASS                     | None                        |
| 000_0000_0000_1000  | 0_0100_0101             | LOCK1_PG1              | FFFF_FFFF                    | PASS                     | None                        |
| 000_0000_0000_1000  | 0_0100_0110             | LOCK2_PG1              | FFFF_FFFF                    | PASS                     | None                        |
| 000_0000_0000_1000  | 0_0100_0111             | LOCK3_PG1              | FFFF_FFFF                    | PASS                     | None                        |
| 000_0000_0000_1000  | 0_0100_1000             | LOCK0_PG2              | FFFF_FFFF                    | PASS                     | None                        |
| 000_0000_0000_1000  | 0_0100_1001             | LOCK1_PG2              | FFFF_FFFF                    | PASS                     | None                        |
| 000_0000_0000_1000  | 0_0100_1010             | LOCK2_PG2              | FFFF_FFFF                    | PASS                     | None                        |
| 000_0000_0000_1000  | 0_0100_1011             | LOCK3_PG2              | FFFF_FFFF                    | PASS                     | None                        |
| 000_0000_0000_1000  | 0_0100_1100             | LOCK0_PG3              | FFFF_FFFF                    | PASS                     | None                        |
| 000_0000_0000_1000  | 0_0100_1101             | LOCK1_PG3              | FFFF_FFFF                    | PASS                     | None                        |
| 000_0000_0000_1000  | 0_0100_1110             | LOCK2_PG3              | FFFF_FFFF                    | PASS                     | None                        |
| 000_0000_0000_1000  | 0_0100_1111             | LOCK3_PG3              | FFFF_FFFF                    | PASS                     | None                        |
| 000_0000_0000_1000  | 0_0101_0000-0_1111_1111 | Reserved               |                              |                          |                             |
| Tamper Detect (TDM) |                         |                        |                              |                          |                             |

Table continues on the next page...

Table 5-6. UTEST DCF clients (continued)

| DCF CS[14:0]       | DCF Address[10:2]       | DCF client description           | DCF client reset value (hex) | DCF client target module | DCF client special strategy |
|--------------------|-------------------------|----------------------------------|------------------------------|--------------------------|-----------------------------|
| 000_0000_0001_0000 | 0_0000_0000             | Diary Base Address               | FFFF_FFFC                    | TDM                      | write_once + wr0_only       |
| 000_0000_0001_0000 | 0_0000_0001             | Tamper Region Override           | 0000_0000                    | TDM                      | wr1_only                    |
| 000_0000_0001_0000 | 0_0000_0010             | Software Region Override Disable | 0000_0000                    | TDM                      | write_once + wr1_only       |
| 000_0000_0001_0000 | 0_0000_0011-0_0000_0111 | Reserved                         |                              |                          |                             |
| 000_0000_0001_0000 | 0_0000_1000             | OTP_EN0                          | 0000_0000                    | TDM                      | wr1_only                    |
| 000_0000_0001_0000 | 0_0000_1001             | OTP_EN1                          | 0000_0000                    | TDM                      | wr1_only                    |
| 000_0000_0001_0000 | 0_0000_1010             | OTP_EN2                          | 0000_0000                    | TDM                      | wr1_only                    |
| 000_0000_0001_0000 | 0_0000_1011             | OTP_EN3                          | 0000_0000                    | TDM                      | wr1_only                    |
| 000_0000_0001_0000 | 0_0000_1100-0_0001_0011 | Reserved                         |                              |                          |                             |
| 000_0000_0001_0000 | 0_0001_0100             | TDR0_LOCK0                       | 0000_0000                    | TDM                      | wr1_only                    |
| 000_0000_0001_0000 | 0_0001_0101             | TDR0_LOCK1                       | 0000_0000                    | TDM                      | wr1_only                    |
| 000_0000_0001_0000 | 0_0001_0110             | TDR0_LOCK2                       | 0000_0000                    | TDM                      | wr1_only                    |
| 000_0000_0001_0000 | 0_0001_0111             | TDR0_LOCK3                       | 0000_0000                    | TDM                      | wr1_only                    |
| 000_0000_0001_0000 | 0_0001_1000             | TDR1_LOCK0                       | 0000_0000                    | TDM                      | wr1_only                    |
| 000_0000_0001_0000 | 0_0001_1001             | TDR1_LOCK1                       | 0000_0000                    | TDM                      | wr1_only                    |
| 000_0000_0001_0000 | 0_0001_1010             | TDR1_LOCK2                       | 0000_0000                    | TDM                      | wr1_only                    |
| 000_0000_0001_0000 | 0_0001_1011             | TDR1_LOCK3                       | 0000_0000                    | TDM                      | wr1_only                    |
| 000_0000_0001_0000 | 0_0001_1100             | TDR2_LOCK0                       | 0000_0000                    | TDM                      | wr1_only                    |
| 000_0000_0001_0000 | 0_0001_1101             | TDR2_LOCK1                       | 0000_0000                    | TDM                      | wr1_only                    |
| 000_0000_0001_0000 | 0_0001_1110             | TDR2_LOCK2                       | 0000_0000                    | TDM                      | wr1_only                    |
| 000_0000_0001_0000 | 0_0001_1111             | TDR2_LOCK3                       | 0000_0000                    | TDM                      | wr1_only                    |
| 000_0000_0001_0000 | 0_0010_0000             | TDR3_LOCK0                       | 0000_0000                    | TDM                      | wr1_only                    |
| 000_0000_0001_0000 | 0_0010_0001             | TDR3_LOCK1                       | 0000_0000                    | TDM                      | wr1_only                    |
| 000_0000_0001_0000 | 0_0010_0010             | TDR3_LOCK2                       | 0000_0000                    | TDM                      | wr1_only                    |
| 000_0000_0001_0000 | 0_0010_0011             | TDR3_LOCK3                       | 0000_0000                    | TDM                      | wr1_only                    |
| 000_0000_0001_0000 | 0_0010_0100             | TDR4_LOCK0                       | 0000_0000                    | TDM                      | wr1_only                    |
| 000_0000_0001_0000 | 0_0010_0101             | TDR4_LOCK1                       | 0000_0000                    | TDM                      | wr1_only                    |
| 000_0000_0001_0000 | 0_0010_0110             | TDR4_LOCK2                       | 0000_0000                    | TDM                      | wr1_only                    |
| 000_0000_0001_0000 | 0_0010_0111             | TDR4_LOCK3                       | 0000_0000                    | TDM                      | wr1_only                    |
| 000_0000_0001_0000 | 0_0010_1000             | TDR5_LOCK0                       | 0000_0000                    | TDM                      | wr1_only                    |
| 000_0000_0001_0000 | 0_0010_1001             | TDR5_LOCK1                       | 0000_0000                    | TDM                      | wr1_only                    |
| 000_0000_0001_0000 | 0_0010_1010             | TDR5_LOCK2                       | 0000_0000                    | TDM                      | wr1_only                    |
| 000_0000_0001_0000 | 0_0010_1011             | TDR5_LOCK3                       | 0000_0000                    | TDM                      | wr1_only                    |
| 000_0000_0001_0000 | 0_0010_1100-0_1111_1111 | Reserved                         |                              |                          |                             |
| MISCELLANEOUS      |                         |                                  |                              |                          |                             |

Table continues on the next page...

Table 5-6. UTEST DCF clients (continued)

| DCF CS[14:0]       | DCF Address[10:2]       | DCF client description                        | DCF client reset value (hex) | DCF client target module | DCF client special strategy        |
|--------------------|-------------------------|-----------------------------------------------|------------------------------|--------------------------|------------------------------------|
| 000_0000_1000_0000 | 0_0000_0000             | UTEST Miscellaneous                           | B008_01C0                    | Several <sup>1</sup>     | Triple Voted <sup>2</sup>          |
| 000_0000_1000_0000 | 0_0000_0001             | UTEST Secure Boot Configuration               | 0000_000D                    | CSE                      | write_once (by NXP)                |
| 000_0000_1000_0000 | 0_0000_0010             | UTEST Secure Boot Code Address                | 0000_0000                    | CSE                      | None                               |
| 000_0000_1000_0000 | 0_0000_0011             | UTEST Secure Boot Code Length                 | 0000_0000                    | CSE                      | None                               |
| 000_0000_1000_0000 | 0_0000_0100             | UTEST XOSC ANA                                | 0000_0004                    | XOSC Analog              | None <sup>2</sup>                  |
| 000_0000_1000_0000 | 0_0000_0101             | UTEST ERROR Functions Config                  | 0000_0000                    |                          | Triple Voted                       |
| 000_0000_1000_0000 | 0_0000_0110             | UTEST Reset Escalation                        | 0000_0000                    | SIU                      | Triple Voted                       |
| 000_0000_1000_0000 | 0_0000_0111-0_0001_1110 | Reserved                                      |                              |                          |                                    |
| 000_0000_1000_0000 | 0_0001_1111             | Reset Event Enable Control                    | 0000_00FF                    | PMC Digital              | Triple Voted + Parity              |
| 000_0000_1000_0000 | 0_0010_0000             | Temperature Sensor Reset Event Enable Control | 0000_003F                    | PMC Digital              | Triple Voted + Parity              |
| 000_0000_1000_0000 | 0_0010_0001             | Reserved                                      |                              |                          |                                    |
| 000_0000_1000_0000 | 0_0010_0010             | VREG Enable                                   | 0000_0000                    | PMC Digital              | Triple Voted + Parity <sup>2</sup> |
| 000_0000_1000_0000 | 0_0010_0011-0_1111_1111 | Reserved                                      |                              |                          |                                    |

1. See Miscellaneous DCF client description

2. Be careful when programming the UTEST Miscellaneous, XOSC ANA, and VREG Enable DCF clients. DCF records are processed sequentially in the order they are read from flash memory. DCF records that control actions such as enabling/disabling voltage regulators or detectors should not be programmed more than once with different values.

### 5.7.1 UTEST Miscellaneous DCF client

This is a 32-bit Triple-Voting DCF Client that implements a variety of settings related to different modules and functionalities, such as boot settings on SIU, oscillator trimming, oscillator enable/disable, and lockstep mode selection. The reset value indicated below is the value when no DCF record has been read or written.

#### NOTE

Be careful when programming the UTEST Miscellaneous, XOSC ANA, and VREG Enable DCF clients. DCF records are processed sequentially in the order they are read from flash

memory. DCF records that control actions such as enabling/disabling voltage regulators or detectors should not be programmed more than once with different values.

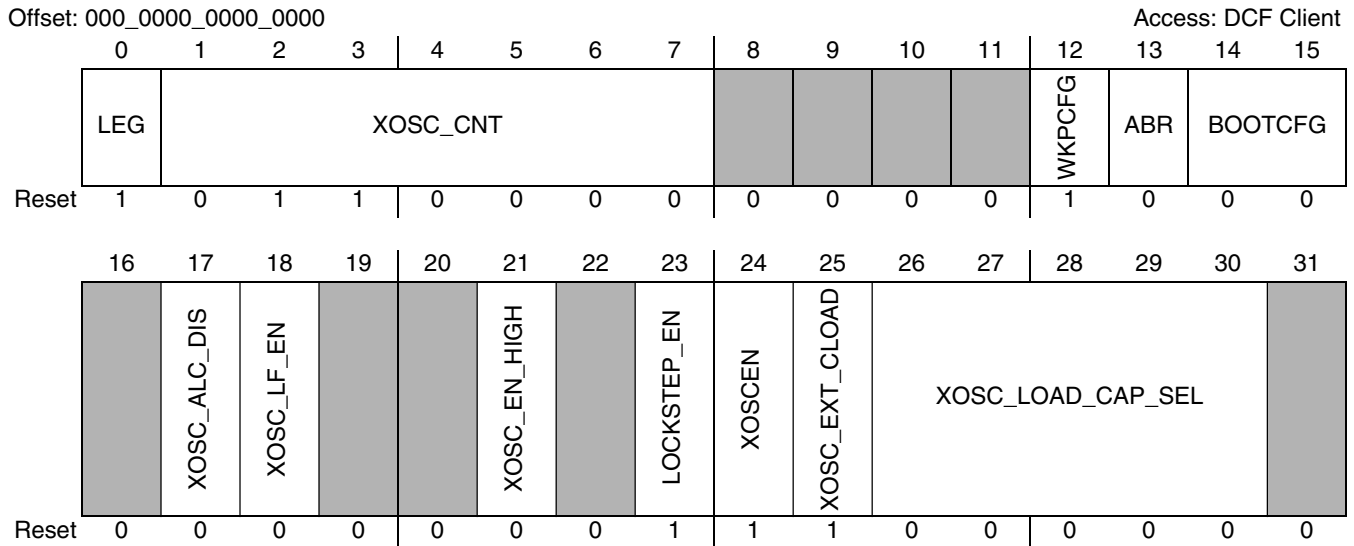


**Figure 5-4. UTEST Miscellaneous DCF client**

**Table 5-7. UTEST Miscellaneous DCF client field descriptions**

| Field            | Description                                                                                                                                                                                                                                  |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>LEG         | Legacy Mode. This bit indicates whether the WKPCFG, ABR, and BOOTCFG settings should come from package pins or from the Miscellaneous DCF client.<br><br>0 Settings come from this DCF client<br>1 Settings driven from pins                 |
| 1-7<br>XOSC_CNT  | Oscillator Stabilization Counter. These seven bits are used to pre-load a counter clocked by the crystal oscillator. Upon reset negation, the counter runs for 512 x XOSC_CNT clock cycles, then sets the XOSC flag in the SIU_RSR register. |
| 8-11<br>Reserved | Reserved                                                                                                                                                                                                                                     |
| 12<br>WKPCFG     | Weak Pull Configuration. SIU_RSR stores this configuration during reset. This information is then used to configure pull up/down of external pins.<br><br>0 Weak pulldown is selected<br>1 Weak pullup is selected                           |
| 13<br>ABR        | Auto Baud Rate. SIU_RSR stores this configuration during reset. This information is then used by the Boot Code to configure the Auto Baud Rate feature.<br><br>0 Auto Baud Rate disabled<br>1 Auto Baud Rate enabled                         |
| 14-15<br>BOOTCFG | Boot Mode Selection. SIU_RSR stores this configuration during reset. This information is then used to by the Boot Code to configure the boot sequence.<br><br>00 Internal boot mode<br>01 Serial boot mode                                   |

*Table continues on the next page...*

**Table 5-7. UTEST Miscellaneous DCF client field descriptions (continued)**

| Field                      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                            | 10 Non-Muxed EBI boot mode (16-bit data, non-muxed address bus)<br>11 Muxed EBI boot mode (16-bit data, muxed address bus)                                                                                                                                                                                                                                                                                                                                                                                   |
| 16<br>Reserved             | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 17<br>XOSC_ALC_DIS         | Oscillator Automatic Level Controller<br>0 Automatic level controller enabled<br>1 Automatic level controller disabled                                                                                                                                                                                                                                                                                                                                                                                       |
| 18<br>XOSC_LF_EN           | Oscillator Low Setting Selection<br>Set this bit to 1 only when XOSC_EN_HIGH is 0. Setting this bit to 1 when XOSC_EN_HIGH is 1 selects a high setting, but this combination is not recommended. <sup>1</sup><br>0 Oscillator is used for medium or high setting<br>1 Oscillator is used for low setting                                                                                                                                                                                                     |
| 19-20<br>Reserved          | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 21<br>XOSC_EN_HIGH         | Oscillator Medium or High Setting Selection<br>If the LEG bit is 0, this field is loaded by the SSCM during reset into the XOSCHIGH bit of SIU_RSR. This bit determines whether the oscillator operates at medium or high setting.<br>Use this bit to select between medium or high setting when XOSC_LF_EN is 0. Setting this bit to 1 when XOSC_LF_EN is 1 selects a high setting, but this combination is not recommended. <sup>1</sup><br>0 Oscillator at medium setting<br>1 Oscillator at high setting |
| 22<br>Reserved             | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 23<br>LOCKSTEP_EN          | Lockstep Mode. This bit is used to enable the Safety Core to be in lockstep with the Core1.<br>0 Lockstep disabled<br>1 Lockstep enabled                                                                                                                                                                                                                                                                                                                                                                     |
| 24<br>XOSCEN               | Crystal Oscillator Enable. This bit is used to enable the crystal oscillator. If disabled, an external clock source can be connected to the EXTAL pin, with the oscillator working in bypass mode.<br>0 Crystal oscillator disabled, bypass mode with clock sourced from EXTAL pin.<br>1 Oscillator enabled, crystal connected.                                                                                                                                                                              |
| 25<br>XOSC_EXT_CLOAD       | Oscillator External Cap<br>0 Selects oscillator internal cap: acceptable for oscillators that require load capacitance of 10 pF or less<br>1 Selects oscillator external cap: required for oscillators that have load capacitance of 10 pF or greater                                                                                                                                                                                                                                                        |
| 26-30<br>XOSC_LOAD_CAP_SEL | Oscillator Load Capacitance Selection. Five bits to program the load capacitance. Load capacitor values are determined by the crystal manufacturer data sheet.                                                                                                                                                                                                                                                                                                                                               |

1. "Low" is the setting commonly used for crystals at 8 MHz, "medium" is commonly used for crystals greater than 8 MHz to 20 MHz, and "high" is commonly used for crystals greater than 20 MHz to 40 MHz. However, the user must characterize

carefully to determine the best  $g_m$  setting for the intended application because crystal load capacitance, board layout, and other factors affect the  $g_m$  value that is needed. See the Data Sheet for additional information.

5.7.2 UTEST Secure Boot Configurations DCF client

This DCF client, in conjunction with the UTEST Secure Boot Code Address and UTEST Secure Boot Code Length clients, configures the CSE security module to carry out a secure boot in the device. This 32-bit write-once DCF client selects:

- whether the CSE module is enabled
- whether the secure boot is to be executed
- the mode (sequential or parallel) in which the secure boot runs
- whether CSE must be disabled in the Failure Analysis (FA) life cycle

The CSE module must be enabled for the secure boot to execute.

When this client selects to execute the secure boot, the boot occurs during reset on the memory space defined by the Secure Boot Code Address and Code Length DCF clients.

NOTE

This write-once DCF client is programmed at the factory with the value 0000\_000Dh. This value cannot be overwritten. This configuration:

- enables CSE
- enables CSE to execute the secure boot during reset
- configures the secure boot to occur in parallel mode
- disables CSE when the life cycle advances to FA

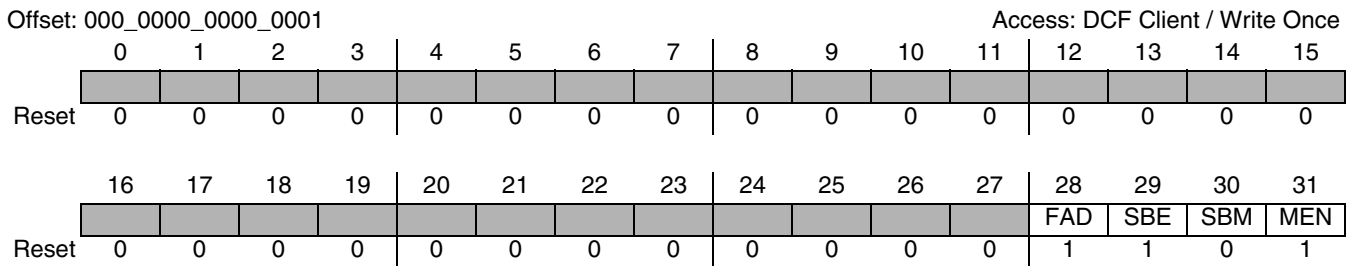


Figure 5-5. UTEST Secure Boot Configurations DCF client

Table 5-8. UTEST Secure Boot Configurations DCF client field descriptions

| Field            | Description    |
|------------------|----------------|
| 0-27<br>Reserved | Reserved       |
| 28<br>FAD        | CSE FA Disable |

Table continues on the next page...



**Table 5-8. UTEST Secure Boot Configurations DCF client field descriptions (continued)**

| Field     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|           | <p>Set this field to 1 to disable the CSE module when the chip's life cycle matures to Failure Analysis (FA). When CSE boots, security information such as the Security Keys is always loaded into RAM. This bit is meant to provide some protection against an MBIST-based attack to extract that security information from the RAM.</p> <p>0 CSE module will not be disabled when device is matured to FA.</p> <p>1 CSE module will be disabled when device is matured to FA.</p> <p><b>NOTE:</b> If the MEN bit is set to 1, the CSE module will be disabled in FA when FAD is 1, but in FA the CSE will still maintain exclusive access to the flash memory blocks mapped to the C55FMC_LOCK1 register, PASS_LOCK1_PGn registers, and TDRn_LOCK1 DCF client.</p> |
| 29<br>SBE | <p>Secure Boot Enable</p> <p>0 Secure boot disabled. CSE will not execute the secure boot during reset.</p> <p>1 Secure boot enabled. CSE will execute the secure boot during reset in parallel or serial mode as per SBM settings.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 30<br>SBM | <p>Secure Boot Mode</p> <p>0 Secure boot runs in parallel mode. In this mode, CSE executes the secure boot in parallel with the user boot code.</p> <p>1 Secure boot runs in sequential mode. In this mode, the user boot mode starts only after CSE finishes the secure boot.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 31<br>MEN | <p>CSE Module Enable</p> <p>This bit indicates that the CSE module will be used in the part.</p> <p>0 CSE is not used. The CSE clock is disabled, and accesses to the registers generate a bus error.</p> <p>1 CSE is enabled.</p> <p><b>NOTE:</b> The secure boot functionality works only if MEN is set to 1. The secure boot functionality does not work if CSE is disabled.</p>                                                                                                                                                                                                                                                                                                                                                                                  |

### 5.7.3 UTEST Secure Boot Code Address DCF client

This DCF client, in conjunction with the UTEST Secure Boot Configurations and UTEST Secure Boot Code Length clients, configures the CSE security module to carry out a secure boot in the device. This 32-bit DCF client provides CSE with the location in memory of the boot code to be verified.

## DCF clients

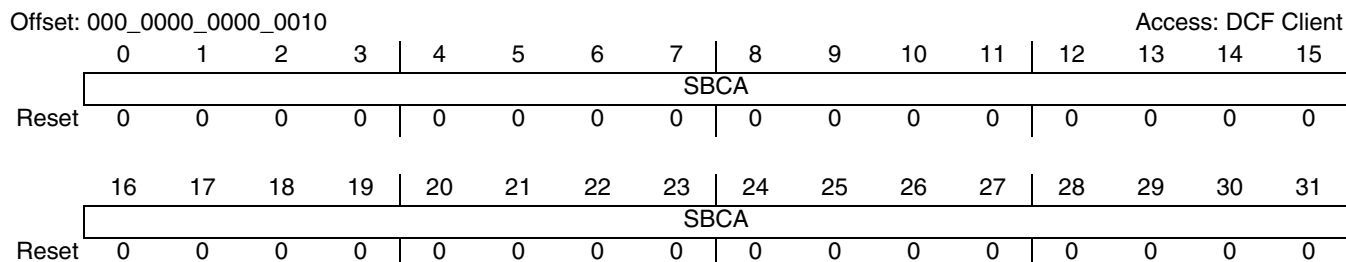


**Figure 5-6. UTEST Secure Boot Code Address DCF client**

**Table 5-9. UTEST Secure Boot Code Address DCF client field descriptions**

| Field | Description                                                            |
|-------|------------------------------------------------------------------------|
| 0-31  | Secure Boot Code Address                                               |
| SBCA  | This 32-bit field indicates the start address of the secure boot code. |

## 5.7.4 UTEST Secure Boot Code Length DCF client

This DCF client, in conjunction with the UTEST Secure Boot Configurations and UTEST Secure Boot Code Address clients, configures the CSE security module to carry out a secure boot in the device. This 32-bit DCF client provides CSE with the length of the secure boot code.

### NOTE

Users whose applications do not use the CSE module, which is automatically enabled, should ensure that the Secure Boot Code Length DCF client has the value 0000\_0000h (its default value).

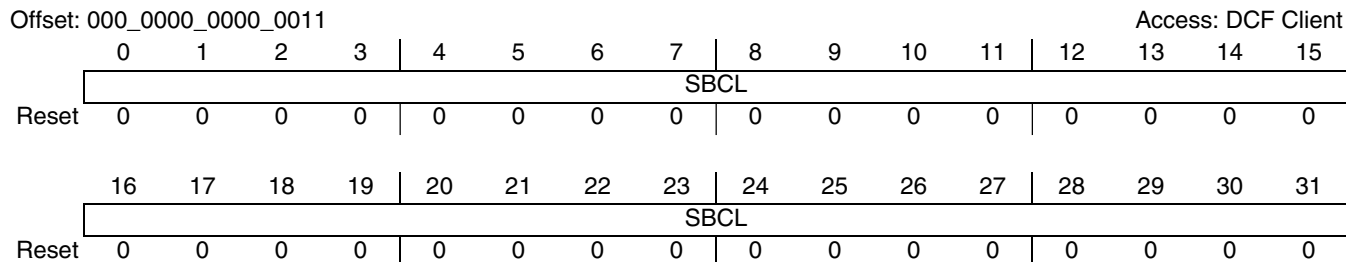


**Figure 5-7. UTEST Secure Boot Code Length DCF client**

**Table 5-10. UTEST Secure Boot Code Length DCF client field descriptions**

| Field | Description                                                            |
|-------|------------------------------------------------------------------------|
| 0-31  | Secure Boot Code Length                                                |
| SBCL  | This 32-bit field indicates the size in bytes of the secure boot code. |

### 5.7.5 UTEST Analog External Oscillator (XOSC ANA) DCF client

This 32-bit DCF client is used to apply settings to the analog external oscillator. The internal current control setting is the only feature available and is used to boot the current. This setting is accessible in functional mode for test and debug purposes.

#### NOTE

Be careful when programming the UTEST Miscellaneous, XOSC ANA, and VREG Enable DCF clients. DCF records are processed sequentially in the order they are read from flash memory. DCF records that control actions such as enabling/disabling voltage regulators or detectors should not be programmed more than once with different values.

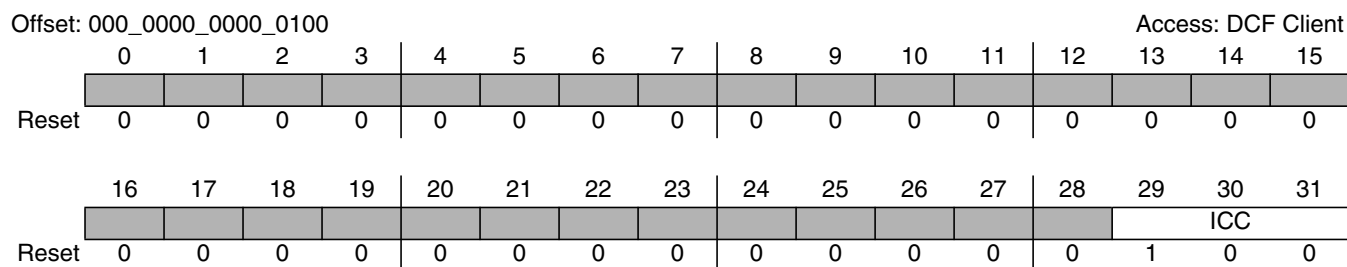


**Figure 5-8. UTEST Analog External Oscillator DCF client**

**Table 5-11. UTEST Analog External Oscillator DCF client field descriptions**

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0-28<br>Reserved | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 29-31<br>ICC     | <p>Internal Current Control</p> <p>These bits allow the internal current to be disabled (no current flowing, so the oscillator does not work), decreased, and increased.</p> <p>000 No (0x) current</p> <p>001 0.25x: decrease current by 25% of default current</p> <p>010 0.5x: decrease current by 50% of default current</p> <p>100 1x: default current, which is 10 mA</p> <p>101 1.25x: increase current by 25% of default current</p> <p>110 1.5x: increase current by 50% of default current</p> <p>111 1.75x: increase current by 75% of default current</p> |

## 5.7.6 UTEST Error Functions Pads Configuration DCF client

This 32-bit Triple-Voting DCF client enables the ERROR function on the ERROR0, ERROR1, and ERRORIN pins independently.

### CAUTION

The ERROR function is disabled by the DCF at power-on reset (POR). After it is enabled, the ERROR function is locked on the pin until the next POR. Any other type of reset has no effect on these settings.

|                            |    |    |    |    |    |    |    |    |    |    |    |                    |      |      |      |     |     |
|----------------------------|----|----|----|----|----|----|----|----|----|----|----|--------------------|------|------|------|-----|-----|
| Offset: 000_0000_0000_0101 |    |    |    |    |    |    |    |    |    |    |    | Access: DCF Client |      |      |      |     |     |
|                            | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11                 | 12   | 13   | 14   | 15  |     |
| Reset                      | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0                  | 0    | 0    | 0    | 0   |     |
|                            | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27                 | 28   | 29   | 30   | 31  |     |
| Reset                      | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0                  | E1OD | E0OD | EINE | E1E | E0E |

**Figure 5-9. UTEST Error Functions Pads Configuration DCF client**

**Table 5-12. UTEST Error Functions Pads Configuration DCF client field descriptions**

| Field            | Description                                                                                                                                                                                      |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0-26<br>Reserved | Reserved                                                                                                                                                                                         |
| 27<br>E1OD       | ERROR1 Open Drain Enable<br>Selects open-drain or push-pull output configuration on ETPUA11 pin. Valid only if E1E is set to 1.<br>0 Push-pull output selection<br>1 Open-drain output selection |
| 28<br>E0OD       | ERROR0 Open Drain Enable<br>Selects open-drain or push-pull output configuration on ETPUA3 pin. Valid only if E0E is set to 1.<br>0 Push-pull output selection<br>1 Open-drain output selection  |
| 29<br>EINE       | ERRORIN Enable<br>Selects multiplexing of ETPUA10 pin.<br>0 ETPUA10 pin function depends on PCR124 settings<br>1 Enables and locks ERRORIN function on ETPUA10 pin                               |
| 30<br>E1E        | ERROR1 Enable<br>Selects multiplexing of ETPUA11 pin.<br>0 ETPUA11 pin function depends on PCR125 settings<br>1 Enables and locks ERROR1 function on ETPUA11 pin                                 |
| 31               | ERROR0 Enable                                                                                                                                                                                    |

**Table 5-12. UTEST Error Functions Pads Configuration DCF client field descriptions**

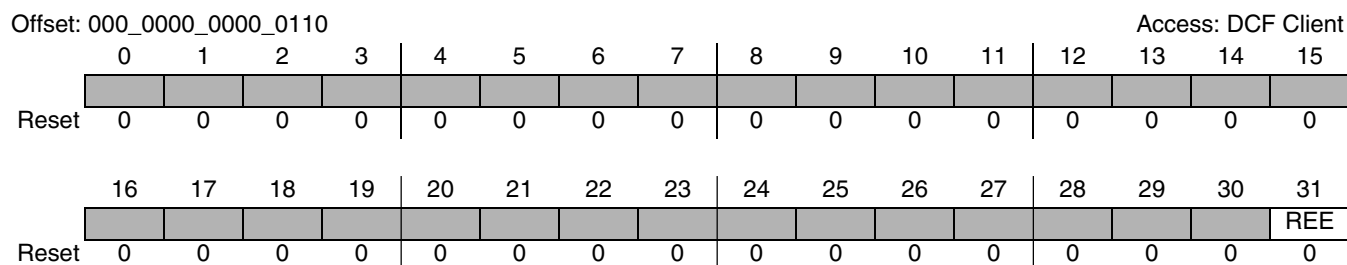
| Field | Description                                                                                                                                  |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------|
| E0E   | Selects multiplexing of ETPUA3 pin.<br>0 ETPUA3 pin function depends on PCR117 settings<br>1 Enables and locks ERROR0 function on ETPUA3 pin |

### 5.7.7 UTEST Reset Escalation Configuration DCF client

This 32-bit Triple-Voting DCF client enables the Reset Escalation function on the reset sequence of the device. See the Reset chapter for further details on this feature.

#### CAUTION

The Reset Escalation function is disabled by the DCF at power-on reset (POR). After it is enabled, the Reset Escalation function is locked until the next POR. Any other type of reset has no effect on these settings.

**Figure 5-10. UTEST Reset Escalation Configuration DCF client****Table 5-13. UTEST Reset Escalation Configuration DCF client field descriptions**

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0-30<br>Reserved | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 31<br>REE        | Reset Escalation Enable<br>Enables a reset escalation counter that is incremented at the end of every reset cycle. If the counter exceeds the number programmed in SIU_RCR[RET], the device is locked into a reset state that can be undone only by a power-up cycle.<br><b>NOTE:</b> LVD/HVD and temperature events that are enabled in the PMC to trigger a reset cause a POR. In these cases, the reset escalation counter resets.<br>0 Reset Escalation disabled<br>1 Reset Escalation enabled |

### 5.7.8 UTEST PMC Reset Enable Control (REE CTRL) DCF client

This DCF client trims the PMC\_REE register. See the PMC chapter for details about this register.

Offset: 0000\_0000\_0001\_1111

### 5.7.9 UTEST PMC Temperature Sensor Reset Event Enable Control (REE TMPSNS) DCF client

This DCF client trims the PMC\_REE\_TD register. See the PMC chapter for details about this register.

Offset: 0000\_0000\_0010\_0000

Access: DCF Client

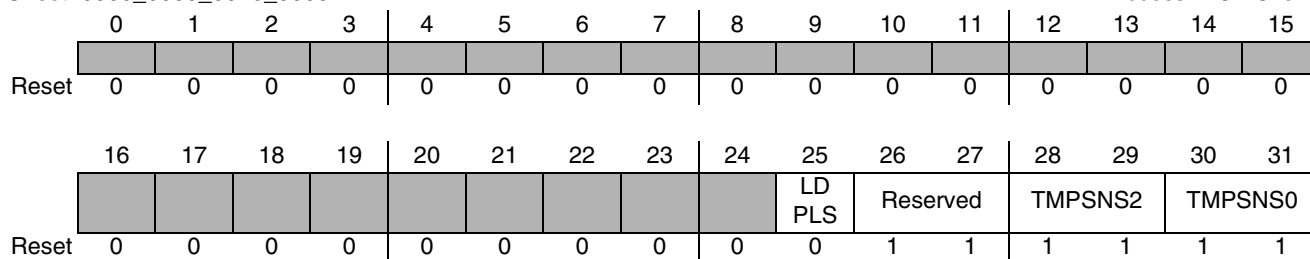


**Figure 5-11. UTEST PMC Temperature Sensor Reset Event Enable Control DCF client**

**Table 5-14. UTEST PMC Temperature Sensor Reset Event Enable Control DCF client field descriptions**

| Field             | Description                                                                                                                                                                |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0-24<br>Reserved  | Reserved                                                                                                                                                                   |
| 25<br>LDPLS       | Load pulse to latch Temperature Sensor REE values<br>0 Load the PMC_REE_TD register with IPS bus data.<br>1 Load the PMC_REE_TD register with the data in this DCF client. |
| 26-27<br>Reserved | Reserved                                                                                                                                                                   |
| 28-29<br>TMPSNS2  | For both Temperature Sensor modules 0 and 1, REE control of temperature 2: 150°C                                                                                           |
| 30-31<br>TMPSNS0  | For both Temperature Sensor modules 0 and 1, REE control of temperature 0: -40°C                                                                                           |

## 5.7.10 UTEST PMC Voltage Regulator Enable (VREG ENB) DCF client

This 32-bit DCF client implements the active low enables for three PMC voltage regulators.

### NOTE

Be careful when programming the UTEST Miscellaneous, XOSC ANA, and VREG Enable DCF clients. DCF records are processed sequentially in the order they are read from flash memory. DCF records that control actions such as enabling/disabling voltage regulators or detectors should not be programmed more than once with different values.

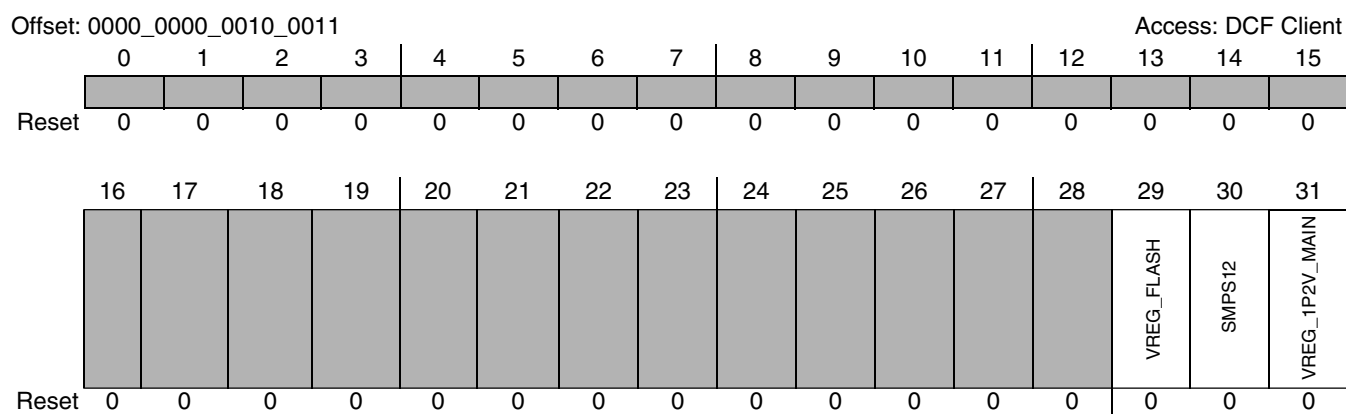


Figure 5-12. UTEST PMC Voltage Regulator Enable DCF client

Table 5-15. UTEST PMC Voltage Regulator Enable DCF client field descriptions

| Field                | Description                                                           |
|----------------------|-----------------------------------------------------------------------|
| 0-28<br>Reserved     | Reserved                                                              |
| 29<br>VREG_FLASH     | Flash memory regulator enable<br>0: Enabled<br>1: Disabled            |
| 30<br>SMPS12         | Switched Mode Power Supply (SMPS) enable<br>0: Enabled<br>1: Disabled |
| 31<br>VREG_1P2V_MAIN | Core regulator enable<br>0: Enabled<br>1: Disabled                    |





## Chapter 6

# e200z7 Core Complex Overview

### 6.1 Core complex

The following table summarizes the cores on this chip.

**Table 6-1. Core complex overview**

| Core identifier | Core type  | Brief description                                                                                             |
|-----------------|------------|---------------------------------------------------------------------------------------------------------------|
| Core0           | e200z759n3 | Main boot computational core                                                                                  |
| Core1           | e200z759n3 | Duplicated computational core                                                                                 |
| Checker core    | e200z758   | Operates in delayed lockstep with Core1, executing exactly the same instructions to confirm correct execution |

### 6.2 Core0 and Core1 overview

The two e200z759n3 cores are identical.

For a comprehensive core description, see the *e200z759n3 Core Reference Manual* (document number e200z759n3CRM).

See [End-to-end Error Correction Code \(e2eECC\)](#) for any limitations of the cores due to the implementation of e2eECC.

### 6.3 Checker core overview

The e200z758 core does not have these features of the e200z759n3 computational cores:

- Instruction cache and data cache
- Nexus 3+ module

Otherwise, the e200z758 checker core has the same features as the e200z759n3 cores.

# Chapter 7

## Platform Coherency Unit (PCU)

### 7.1 Introduction

The Platform Coherency Unit is a hardware mechanism for maintaining memory coherency between multiple bus masters. It supports cache coherency between the two cores, including writes via the Nexus read-write access block. In addition, DMA writes to system memory may be configured to provide cache coherency.

For writes originating from either core, the MMU for the core must be configured with Memory Coherence (M bit) and Write Through (W bit) set for the address spaces to be accessed. For writes originating from either DMA module, use the [Global PCU Output Register \(DMA\\_GPOR<sub>n</sub>\)](#) to enable coherency on a channel-by-channel basis. Finally, set the Master Write Monitor Enable bits for each master, requiring coherency for its writes, and the global enable bit (ENB) in the PCU's configuration register.

#### NOTE

On predecessor devices, the Platform Coherency Unit (PCU) was named the Cache Coherency Unit (CCU).

### 7.2 Memory map and register descriptions

The programming model contains control and status registers. It can be referenced using only 32-bit (word) accesses. Writes to the configuration fields can be performed only in supervisor mode. Attempted accesses using non-32-bit data sizes, accesses to undefined or reserved addresses, or accesses with a non-supported access type result in a bus error termination.

## PCU memory map

| Address offset (hex) | Register name                                          | Width (in bits) | Access | Reset value | Section/ page             |
|----------------------|--------------------------------------------------------|-----------------|--------|-------------|---------------------------|
| 0                    | PCU Configuration and Error Status Register (PCU_CESR) | 32              | R/W    | 0300_0000h  | <a href="#">7.2.1/152</a> |
| 10                   | PCU Error Address Register, Core n (PCU_EAR0)          | 32              | R      | 0000_0000h  | <a href="#">7.2.2/156</a> |
| 14                   | PCU Error Detail Register, Core n (PCU_EDR0)           | 32              | R      | 0000_0000h  | <a href="#">7.2.3/157</a> |
| 20                   | PCU Error Address Register, Core n (PCU_EAR1)          | 32              | R      | 0000_0000h  | <a href="#">7.2.2/156</a> |
| 24                   | PCU Error Detail Register, Core n (PCU_EDR1)           | 32              | R      | 0000_0000h  | <a href="#">7.2.3/157</a> |
| 30                   | PCU Interrupt Register, Core n (PCU_IR0)               | 32              | R/W    | 0000_0000h  | <a href="#">7.2.4/158</a> |
| 34                   | PCU Interrupt Register, Core n (PCU_IR1)               | 32              | R/W    | 0000_0000h  | <a href="#">7.2.4/158</a> |

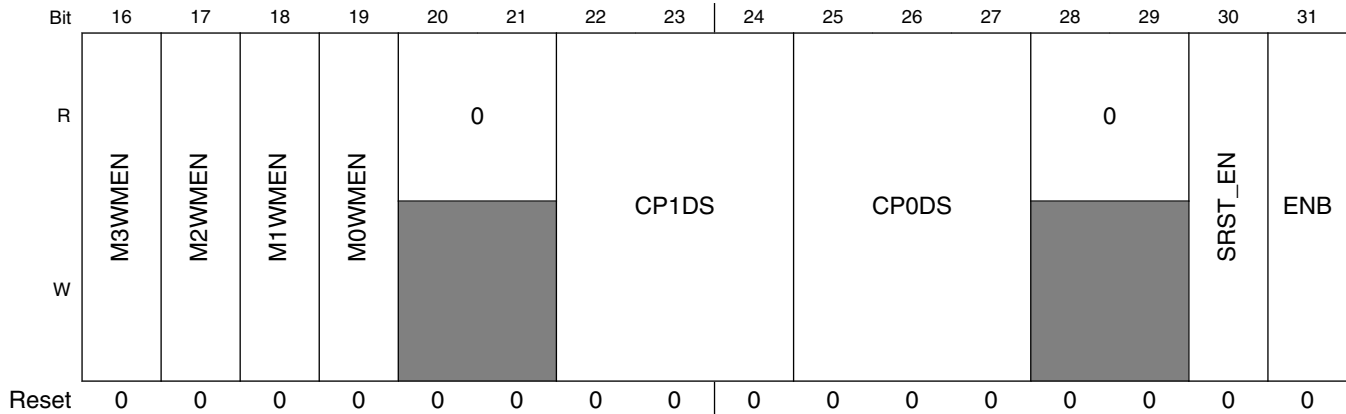
## 7.2.1 PCU Configuration and Error Status Register (PCU\_CESR)

Access by mode:

- Read: all modes
- Write: Supervisor mode

Address: 0h base + 0h offset = 0h

| Bit   | 0           | 1 | 2           | 3 | 4         | 5 | 6       | 7       | 8 | 9 | 10     | 11     | 12 | 13 | 14 | 15 |
|-------|-------------|---|-------------|---|-----------|---|---------|---------|---|---|--------|--------|----|----|----|----|
| R     | CP1ERR[1:0] |   | CP0ERR[1:0] |   | SRST_FLAG | 0 | CP1IDLE | CP0IDLE | 0 |   | CP1IEN | CP0IEN | 0  |    |    |    |
| W     | w1c         |   | w1c         |   |           |   |         |         |   |   |        |        |    |    |    |    |
| Reset | 0           | 0 | 0           | 0 | 0         | 0 | 1       | 1       | 0 | 0 | 0      | 0      | 0  | 0  | 0  | 0  |



PCU\_CESR field descriptions

| Field              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–1<br>CP1ERR[1:0] | <p>Core Processor 1 Error</p> <p>This field represents a flag maintained by the PCU for signaling the presence of a captured snoop error or the occurrence of a queue overflow. The error details are contained in PCU_EAR1 and PCU_EDR1. CP1ERR[1] is set when the snoop queue for Core1 has overflowed; the address and attributes of the global write that caused the queue overflow are recorded in PCU_EAR1 and PCU_EDR1. CP1ERR[0] is set when the hardware detects a snoop error and records the faulting address and attributes. This field is cleared when the corresponding bits are written as 1. If another Core1 error occurs before the CP1ERR bit is cleared, CP1ERR remains set while PCU_EAR1 and PCU_EDR1 capture address and attributes for the most recent error event.</p> <p>00 PCU_EAR1/PCU_EDR1 do not contain a captured snoop error.<br/> 01 PCU_EAR1/PCU_EDR1 contain a captured snoop error.<br/> 10 PCU_EAR1/PCU_EDR1 contain a queue overflow occurrence.<br/> 11 A snoop error event occurred, followed by a queue overflow event. Because the PCU_EAR1/PCU_EDR1 registers capture the most recent error event, PCU_EAR1/PCU_EDR1 contain the queue overflow occurrence.</p> |
| 2–3<br>CP0ERR[1:0] | <p>Core Processor 0 Error</p> <p>This field represents a flag maintained by the PCU for signaling the presence of a captured snoop error or the occurrence of a queue overflow. The error details are contained in PCU_EAR0 and PCU_EDR0. CP0ERR[1] is set when the snoop queue for Core0 has overflowed; the address and attributes of the global write that caused the queue overflow are recorded in PCU_EAR0 and PCU_EDR0. CP0ERR[0] is set when the hardware detects a snoop error and records the faulting address and attributes. This field is cleared when the corresponding bits are written as 1. If another Core0 error occurs before the CP0ERR bit is cleared, CP0ERR remains set while PCU_EAR0 and PCU_EDR0 capture address and attributes for the most recent error event.</p> <p>00 PCU_EAR0/PCU_EDR0 do not contain a captured snoop error.<br/> 01 PCU_EAR0/PCU_EDR0 contain a captured snoop error.<br/> 10 PCU_EAR0/PCU_EDR0 contain a queue overflow occurrence.<br/> 11 A snoop error event occurred, followed by a queue overflow event. Because the PCU_EAR0/PCU_EDR0 registers capture the most recent error event, PCU_EAR0/PCU_EDR0 contain the queue overflow occurrence.</p> |
| 4<br>SRST_FLAG     | <p>Software Reset Flag</p> <p>This field represents a flag maintained by the PCU to indicate that a software reset event has occurred. A software reset of the PCU is invoked by programming the PCU_CESR[SRST_EN] field.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |

Table continues on the next page...

**PCU\_CESR field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                          |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | 0 The last PCU reset event was a hardware reset event.<br>1 The last PCU reset event was a software-invoked reset event.                                                                                                                                                                                                                                             |
| 5<br>Reserved     | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                              |
| 6<br>CP1IDLE      | Core Processor 1 Idle<br>This read-only status flag indicates when the Core1 snoop queue is empty.<br>0 Core1 snoop queue is not empty; Core1 snoop queue is busy processing valid snoop requests to Core1.<br>1 Core1 snoop queue is empty.                                                                                                                         |
| 7<br>CP0IDLE      | Core Processor 0 Idle<br>This read-only status flag indicates when the Core0 snoop queue is empty.<br>0 Core0 snoop queue is not empty; Core0 snoop queue is busy processing valid snoop requests to Core0.<br>1 Core0 snoop queue is empty.                                                                                                                         |
| 8–9<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                              |
| 10<br>CP1IEN      | Core Processor 1 Interrupt Enable<br>This bit enables interrupt generation upon detection of a Core1 snoop request that results in an error.<br>0 PCU generates an interrupt request upon detection of a Core1 snoop that terminated with error.<br>1 PCU does not generate an interrupt request upon detection of a Core1 snoop request that terminated with error. |
| 11<br>CP0IEN      | Core Processor 0 Interrupt Enable<br>This bit enables interrupt generation upon detection of a Core0 snoop request that results in an error.<br>0 PCU generates an interrupt request upon detection of a Core0 snoop that terminated with error.<br>1 PCU does not generate an interrupt request upon detection of a Core0 snoop request that terminated with error. |
| 12–15<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                              |
| 16<br>M3WMEN      | Master 3 Write Monitor Enable<br>This bit enables monitoring of eDMA_B for global writes.<br>0 Disable eDMA_B monitoring. Global writes from eDMA_B do not initiate any snoop requests.<br>1 Enable eDMA_B monitoring for global writes.                                                                                                                             |
| 17<br>M2WMEN      | Master 2 Write Monitor Enable<br>This bit enables monitoring of eDMA_A for global writes.<br>0 Disable eDMA_A monitoring. Global writes from eDMA_A do not initiate any snoop requests.<br>1 Enable eDMA_A monitoring for global writes.                                                                                                                             |
| 18<br>M1WMEN      | Master 1 Write Monitor Enable<br>This bit enables monitoring of Core1 for global writes.                                                                                                                                                                                                                                                                             |

*Table continues on the next page...*

**PCU\_CESR field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | 0 Disable Core1 monitoring. Global writes from Core1 do not initiate any snoop requests.<br>1 Enable Core1 monitoring for global writes.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 19<br>M0WMEN      | Master 0 Write Monitor Enable<br><br>This bit enables monitoring of Core0 for global writes.<br><br>0 Disable Core0 monitoring. Global writes from Core0 do not initiate any snoop requests.<br>1 Enable Core0 monitoring for global writes.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 20–21<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 22–24<br>CP1DS    | Core1 Disable Snoop<br><br>This field allows the suspension of snooping activity targeting Core1. Snooping can be disabled universally for Core1, upon detection of a Core1 snoop termination with error or queue overflow occurrence, or upon detection of an overflow of the Core0 snoop queue.<br><br>000 Core1 snooping disabled upon detection of an overflow of the Core1 snoop queue, as flagged in PCU_CESR[CP1ERR]. When PCU_CESR[CP1ERR] is cleared, Core1 snooping activity can resume.<br>001 Core1 snooping disabled upon detection of an overflow of the Core1 snoop queue or upon a Core1 snoop termination with error, as flagged in PCU_CESR[CP1ERR]. Any valid entries in the Core1 snoop queue will be invalidated and will not be processed. When PCU_CESR[CP1ERR] is cleared, Core1 snooping activity can resume.<br>010 Core1 snooping disabled upon detection of an overflow of the Core1 snoop queue or upon detection of an overflow of the Core0 snoop queue, as flagged in PCU_CESR[CP1ERR] or PCU_CESR[CP0ERR]. Any valid entries in the Core1 snoop queue will be invalidated and will not be processed. When PCU_CESR[CP1ERR] or PCU_CESR[CP0ERR] (as applicable) is cleared, Core1 snooping activity can resume.<br>011 Core1 snooping disabled upon a Core1 snoop termination with error or upon detection of an overflow of either the Core1 snoop queue or the Core0 snoop queue, as flagged in PCU_CESR[CP1ERR] or PCU_CESR[CP0ERR]. Any valid entries in the Core1 snoop queue will be invalidated and will not be processed. When PCU_CESR[CP1ERR] or PCU_CESR[CP0ERR] (as applicable) is cleared, Core1 snooping activity can resume.<br>1xx Core1 snooping entirely disabled. Any valid entries in the Core1 snoop queue will be invalidated and will not be processed. |
| 25–27<br>CP0DS    | Core0 Disable Snoop<br><br>This field allows the suspension of snooping activity targeting Core0. Snooping can be disabled universally for Core0, upon detection of a Core0 snoop termination with error or queue overflow occurrence, or upon detection of an overflow of the Core1 snoop queue.<br><br>000 Core0 snooping disabled upon detection of an overflow of the Core0 snoop queue, as flagged in PCU_CESR[CP0ERR]. When PCU_CESR[CP0ERR] is cleared, Core0 snooping activity can resume.<br>001 Core0 snooping disabled upon detection of an overflow of the Core0 snoop queue or upon a Core0 snoop termination with error, as flagged in PCU_CESR[CP0ERR]. Any valid entries in the Core0 snoop queue will be invalidated and will not be processed. When PCU_CESR[CP0ERR] is cleared, Core0 snooping activity can resume.<br>010 Core0 snooping disabled upon detection of an overflow of the Core0 snoop queue or upon detection of an overflow of the Core1 snoop queue, as flagged in PCU_CESR[CP0ERR] or PCU_CESR[CP1ERR]. Any valid entries in the Core0 snoop queue will be invalidated and will not be processed. When PCU_CESR[CP0ERR] or PCU_CESR[CP1ERR] (as applicable) is cleared, Core0 snooping activity can resume.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

*Table continues on the next page...*

**PCU\_CESR field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 011               | Core0 snooping disabled upon a Core0 snoop termination with error or upon detection of an overflow of either the Core0 snoop queue or the Core1 snoop queue, as flagged in PCU_CESR[CP0ERR] or PCU_CESR[CP1ERR]. Any valid entries in the Core0 snoop queue will be invalidated and will not be processed. When PCU_CESR[CP0ERR] or PCU_CESR[CP1ERR] (as applicable) is cleared, Core0 snooping activity can resume.                                                                                                                                                                                            |
| 1xx               | Core0 snooping entirely disabled. Any valid entries in the Core0 snoop queue will be invalidated and will not be processed.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 28–29<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 30<br>SRST_EN     | Software Reset Enable<br><br>Writing 1 to this bit invokes a software-driven reset of the PCU. On the next cycle following the programming of PCU_CESR[SRST_EN], all programming model registers in the PCU are reset, and all snoop queue entries are invalidated. Then PCU_CESR[SRST_FLAG] is set to 1 to indicate that a software-driven reset of the PCU has occurred.<br><br>Writing 0 to this bit has no effect. Reading this bit returns 0.<br><br>0 No software-driven reset is requested. Writing 0 to this field has no effect. Reading this bit always returns 0.<br>1 Invoke software-driven reset. |
| 31<br>ENB         | PCU Enable<br><br>This bit provides a global enable/disable for the PCU.<br><br>While the PCU is disabled, hardware coherency checking is not performed.<br><br>0 The PCU is disabled.<br>1 The PCU is enabled.                                                                                                                                                                                                                                                                                                                                                                                                 |

**7.2.2 PCU Error Address Register, Core n (PCU\_EAR<sub>n</sub>)**

This register can be read in all modes.

Address: 0h base + 10h offset + (16d × i), where i=0d to 1d

|       |       |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0     | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | EADDR |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |       |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0     | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**PCU\_EAR<sub>n</sub> field descriptions**

| Field         | Description                                                                                                                                           |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>EADDR | Error Address<br><br>This read-only field is the reference address associated with the global write that resulted in a snoop error or overflow event. |



### 7.2.3 PCU Error Detail Register, Core n (PCU\_EDRn)

This register can be read in all modes.

Address: 0h base + 14h offset + (16d × i), where i=0d to 1d

|       |       |   |   |   |      |   |   |   |        |   |    |    |    |    |    |    |    |    |       |    |    |    |       |    |    |    |    |    |    |    |    |    |
|-------|-------|---|---|---|------|---|---|---|--------|---|----|----|----|----|----|----|----|----|-------|----|----|----|-------|----|----|----|----|----|----|----|----|----|
| Bit   | 0     | 1 | 2 | 3 | 4    | 5 | 6 | 7 | 8      | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18    | 19 | 20 | 21 | 22    | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | EMSTR |   |   |   | EPTR |   |   |   | ESTAMP |   |    |    |    |    |    |    | 0  |    | EPROT |    |    |    | ERESP |    |    |    | 0  |    |    |    |    |    |
| W     |       |   |   |   |      |   |   |   |        |   |    |    |    |    |    |    |    |    |       |    |    |    |       |    |    |    |    |    |    |    |    |    |
| Reset | 0     | 0 | 0 | 0 | 0    | 0 | 0 | 0 | 0      | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0     | 0  | 0  | 0  | 0     | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

#### PCU\_EDRn field descriptions

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–3<br>EMSTR      | <p>Error Master</p> <p>This read-only field holds the AHB master ID associated with the global write that resulted in a snoop error or overflow event.</p>                                                                                                                                                                                                                                                                                                                                                                                          |
| 4–7<br>EPTR       | <p>Error Pointer</p> <p>This read-only field holds the value of the queue pointer associated with the global write that resulted in a snoop error or overflow event.</p> <p>Upon a snoop error event, this field describes the relative position of the corresponding entry in the 16-entry queue for the global write that resulted in a snoop error.</p> <p>Upon a queue overflow event, this field describes the relative position of the corresponding entry in the 16-entry queue for the global write that resulted in an overflow event.</p> |
| 8–15<br>ESTAMP    | <p>Error Time Stamp</p> <p>This read-only field holds the time stamp value associated with the snoop error or overflow event. The time stamp value is based on a free-running 8-bit timer that begins counting upon assertion of PCU_CESR[ENB].</p> <p>Upon a snoop error event, this field captures the value of the time stamp at the time the snoop request was loaded into the queue.</p> <p>Upon a queue overflow event, this field captures the value of the time stamp at the time the overflow condition is detected.</p>                   |
| 16–17<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 18–23<br>EPROT    | <p>Error Protection Attribute</p> <p>This read-only field holds the AHB protection attributes that resulted in a snoop error or overflow event.</p>                                                                                                                                                                                                                                                                                                                                                                                                 |
| 24–28<br>ERESP    | <p>Error Snoop Response</p> <p>This read-only field holds the snoop response associated with the snoop that terminated with error. This field is cleared when PCU_EDRn is loaded on an overflow event. Upon a queue overflow event, this field reads as all zeroes.</p> <p>In the field-setting descriptions, cc designates the number of collapsed requests:</p> <ul style="list-style-type: none"> <li>00: no collapsing</li> <li>01: two requests combined</li> <li>10: three requests combined</li> <li>11: four requests combined</li> </ul>   |

Table continues on the next page...

**PCU\_EDR<sub>n</sub> field descriptions (continued)**

| Field             | Description                                                                             |
|-------------------|-----------------------------------------------------------------------------------------|
| 000cc             | NULL: no operation performed or no matching cache entry                                 |
| 001cc             | AutoInv: AutoInvalidation performed on clean unlocked lines with tag parity errors      |
| 010cc             | ERROR: Error in processing a snoop request due to tag parity error                      |
| 011cc             | SYNC: Sync completed, snoop queue synchronized                                          |
| 100cc             | Hit Clean: matching unlocked clean cache entry found                                    |
| 101cc             | Hit Dirty: matching unlocked dirty cache entry found                                    |
| 110cc             | Hit Locked: matching locked clean cache entry found                                     |
| 111cc             | Hit Dirty Locked: matching locked clean cache entry found                               |
| 29–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |

**7.2.4 PCU Interrupt Register, Core n (PCU\_IR<sub>n</sub>)**

Access by mode:

- Read: all modes
- Write: Supervisor mode

Address: 0h base + 30h offset + (4d × i), where i=0d to 1d

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |     |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|-----|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14  | 15  |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |     |     |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |     |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0   |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30  | 31  |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    | OIF | EIF |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    | w1c | w1c |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0   |

**PCU\_IR<sub>n</sub> field descriptions**

| Field            | Description                                                                                                                                                                                                                                                                                                      |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–29<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                          |
| 30<br>OIF        | Overflow Interrupt Flag<br><br>This field indicates an outstanding interrupt request as a result of a queue overflow. The flag and interrupt are cleared by writing 1 to this bit. Writing 0 to this bit has no effect.<br><br>0 No interrupt request<br>1 Interrupt request due to snoop termination with error |
| 31<br>EIF        | Error Interrupt Flag                                                                                                                                                                                                                                                                                             |

*Table continues on the next page...*

**PCU\_IR<sub>n</sub> field descriptions (continued)**

| Field | Description                                                                                                                                                                                                                    |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | This field indicates an outstanding interrupt request as a result of a snoop that terminated with error if PCU_CESR[CPnIEN]. The flag and interrupt are cleared by writing 1 to this bit. Writing 0 to this bit has no effect. |
| 0     | No interrupt request                                                                                                                                                                                                           |
| 1     | Interrupt request due to snoop termination with error                                                                                                                                                                          |



## Chapter 8

# System Integration Unit (SIU, SIU\_B)

### 8.1 Introduction

This chapter describes the device system integration units (SIU, and SIU\_B) that configure and initialize the following features in the SIU:

- MCU reset configuration
- System reset operation
- Pad configuration
- External interrupts
- General-purpose I/O (GPIO)
- Internal peripheral multiplexing
- GPDI and GPDO I/Os of the DSPI modules
- TLB Entry Selector

as well as the following feature in SIU\_B:

- Mirrored pad configuration and GPIO registers for Protected Port Output mode

#### 8.1.1 Block diagram

The figure below is a block diagram of the SIU. The Power-on Reset Detection, Pad Ring/Pad Interface, and the Peripheral I/O Channels are external to the SIU.

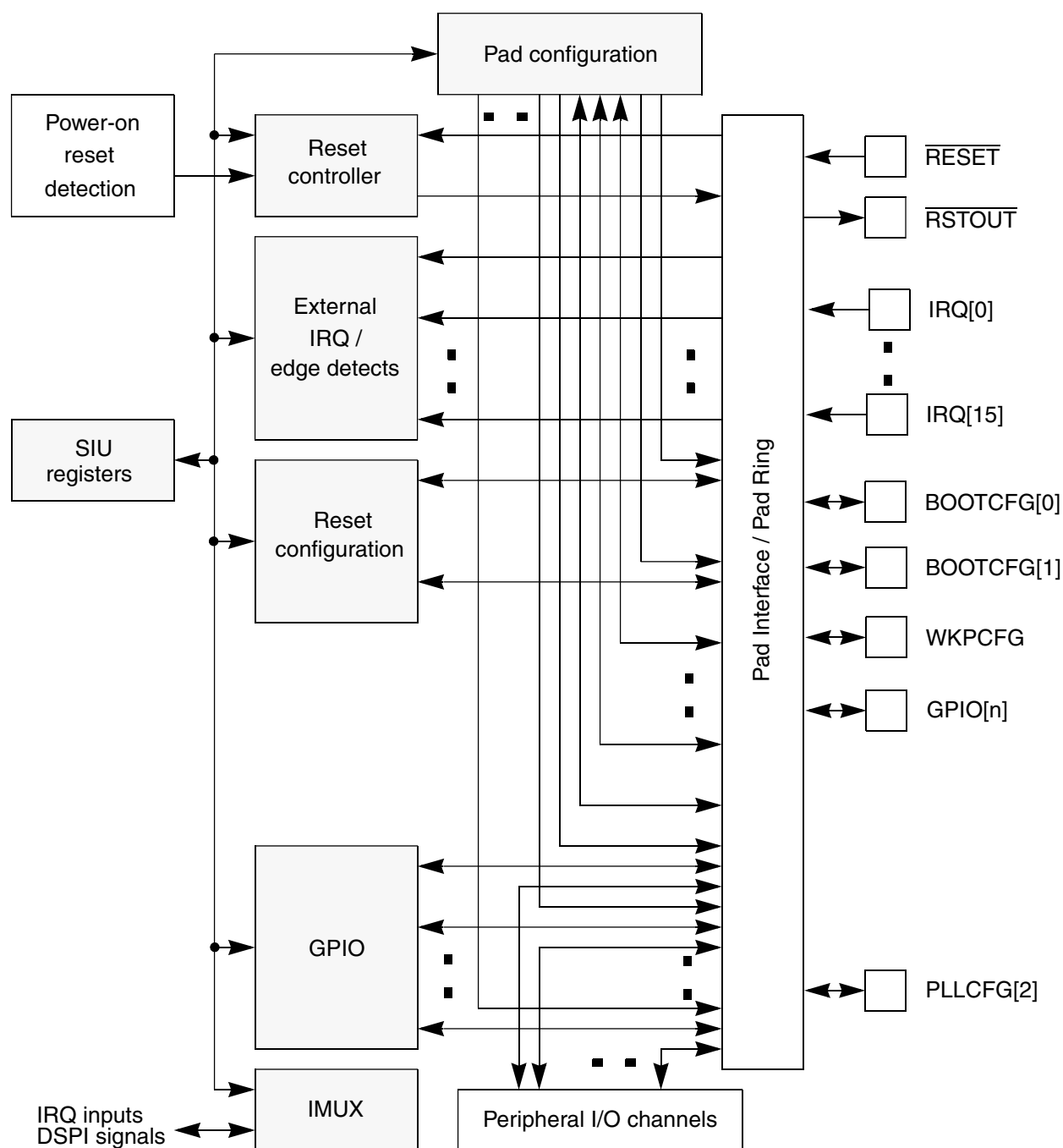


Figure 8-1. SIU block diagram

## 8.1.2 Overview

The following table lists the features that the SIU configures.

**Table 8-1. SIU features**

| Feature                          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|----------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MCU reset operations             | Controls the external pin boot logic                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| System reset operations          | Monitors internal and external reset sources, and drives the $\overline{\text{RSTOUT}}$ signal <ul style="list-style-type: none"> <li>• Power-on reset support</li> <li>• Reset status register providing last reset source to software</li> <li>• Glitch detection on reset input</li> <li>• Software controlled reset assertion</li> </ul>                                                                                                                                            |
| Pad configuration registers      | Enables the configuration and initialization of the I/O pin electrical characteristics using software to select the following: <ul style="list-style-type: none"> <li>• Active function from the set of multiplexed functions</li> <li>• Pullup and pulldown characteristics of the pin</li> <li>• Slew rate for slow and medium pads</li> <li>• Open drain mode for output pins</li> <li>• Hysteresis for input pins</li> <li>• Drive strength of bus signals for fast pads</li> </ul> |
| External interrupt operations    | 16 interrupt requests<br>Rising- or falling-edge event detection<br>Programmable digital filter for glitch rejection<br>NMI and critical interrupt control                                                                                                                                                                                                                                                                                                                              |
| General purpose I/O (GPIO)       | Provides uniform and discrete I/O control of MCU general-purpose I/O pins, where each GPIO signal has an input register and an output register.                                                                                                                                                                                                                                                                                                                                         |
| Internal peripheral multiplexing | Provides flexibility to customize signal/pin assignments for application development that allows: <ul style="list-style-type: none"> <li>• Flexible selection of eQADC trigger inputs</li> <li>• Assignment of interrupt requests (IRQs) between external pins and DSPI</li> </ul>                                                                                                                                                                                                      |
| TLB entry selector               | Allows an external tool to non-intrusively change the TLB entry that is used by the cores to access memory.                                                                                                                                                                                                                                                                                                                                                                             |
| Protected Port Output            | Allows up to four GPIO pins to be reserved by any one of the two cores.                                                                                                                                                                                                                                                                                                                                                                                                                 |

## 8.1.3 Modes of operation

### 8.1.3.1 Normal mode

In normal mode, the SIU provides the register interface and logic that controls the device and system configuration, the reset controller, and GPIO.

### 8.1.3.2 Debug mode

SIU operation in debug mode is identical to operation in normal mode.

## 8.2 Memory Map and Register Definition

Register offsets are relative to SIU's base address.

### Pad Configuration Register (PCR) details

[Pad Configuration Register \(SIU\\_PCR<sub>n</sub>\)](#) provides a general description of the PCRs. The following tables provide details about individual PCRs. Gaps exist in the memory map where registers correspond to GPIO pins that are not implemented in the package.

[Table 8-2](#) defines the symbols for PCR<sub>n</sub> fields' reset values that appear in [Table 8-3](#).

**Table 8-2. SIU\_PCR<sub>n</sub> field reset-value symbols**

| Symbol | Meaning                                                                                                                                                 |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| —      | Not implemented in the PCR. Software should write a default value of 0.                                                                                 |
| 0 or 1 | Function is implemented. For its definition, see the applicable field's description.                                                                    |
| U      | Function is implemented and is defined by the WKPCFG bit of the Miscellaneous DCF client (or, in legacy mode, by the value of the WKPCFG pin at reset). |

**Table 8-3. SIU\_PCR<sub>n</sub> settings**

| P<br>C<br>R     | Offset<br>(hex) | GPIO   | Primary<br>function | Alternate 1       | Alternate 2          | Alternate 3 | SIU_PCR <sub>n</sub> [3:15]<br>(SIU_PCR <sub>n</sub> [0:2] = reserved) |             |             |             |             |                  |                  |                       |                       |                       |                       |                       |                       |
|-----------------|-----------------|--------|---------------------|-------------------|----------------------|-------------|------------------------------------------------------------------------|-------------|-------------|-------------|-------------|------------------|------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
|                 |                 |        |                     |                   |                      |             | 3                                                                      | 4           | 5           | 6           | 7           | 8                | 9                | 10                    | 11                    | 12                    | 13                    | 14                    | 15                    |
|                 |                 |        |                     |                   |                      |             | P<br>A<br>2                                                            | P<br>A<br>1 | P<br>A<br>0 | O<br>B<br>E | I<br>B<br>E | D<br>S<br>C<br>1 | D<br>S<br>C<br>0 | O<br>D<br>Y<br>E<br>S | H<br>Y<br>R<br>C<br>1 | S<br>R<br>P<br>C<br>0 | S<br>R<br>P<br>C<br>1 | W<br>P<br>P<br>E<br>S | W<br>P<br>P<br>E<br>S |
| 75 <sup>1</sup> | 00D6            | GPIO75 | MDO4                |                   |                      |             | —                                                                      | —           | —           | 0           | 0           | —                | —                | 0                     | 0                     | 1                     | 1                     | 1                     | 0                     |
| 76 <sup>1</sup> | 00D8            | GPIO76 | MDO5                |                   |                      |             | —                                                                      | —           | —           | 0           | 0           | —                | —                | 0                     | 0                     | 1                     | 1                     | 1                     | 0                     |
| 77 <sup>1</sup> | 00DA            | GPIO77 | MDO6                |                   |                      |             | —                                                                      | —           | —           | 0           | 0           | —                | —                | 0                     | 0                     | 1                     | 1                     | 1                     | 0                     |
| 78 <sup>1</sup> | 00DC            | GPIO78 | MDO7                |                   |                      |             | —                                                                      | —           | —           | 0           | 0           | —                | —                | 0                     | 0                     | 1                     | 1                     | 1                     | 0                     |
| 79 <sup>1</sup> | 00DE            | GPIO79 | MDO8                |                   |                      |             | —                                                                      | —           | —           | 0           | 0           | —                | —                | 0                     | 0                     | 1                     | 1                     | 1                     | 0                     |
| 80 <sup>1</sup> | 00E0            | GPIO80 | MDO9                |                   |                      |             | —                                                                      | —           | —           | 0           | 0           | —                | —                | 0                     | 0                     | 1                     | 1                     | 1                     | 0                     |
| 81 <sup>1</sup> | 00E2            | GPIO81 | MDO10               |                   |                      |             | —                                                                      | —           | —           | 0           | 0           | —                | —                | 0                     | 0                     | 1                     | 1                     | 1                     | 0                     |
| 82 <sup>1</sup> | 00E4            | GPIO82 | MDO11               |                   |                      |             | —                                                                      | —           | —           | 0           | 0           | —                | —                | 0                     | 0                     | 1                     | 1                     | 1                     | 0                     |
| 83              | 00E6            | GPIO83 | CNTXA               | TXDA <sup>2</sup> | MCANTX0              |             | —                                                                      | 0           | 0           | 0           | 0           | —                | —                | 0                     | 0                     | 0                     | 0                     | 1                     | 1                     |
| 84              | 00E8            | GPIO84 | CNRXA               | RXDA <sup>2</sup> | MCANRX0 <sup>2</sup> |             | —                                                                      | 0           | 0           | 0           | 0           | —                | —                | 0                     | 0                     | 0                     | 0                     | 1                     | 1                     |
| 85              | 00EA            | GPIO85 | CNTXB               | PCSC3             |                      |             | —                                                                      | 0           | 0           | 0           | 0           | —                | —                | 0                     | 0                     | 0                     | 0                     | 1                     | 1                     |
| 86              | 00EC            | GPIO86 | CNRXB               | PCSC4             |                      |             | —                                                                      | 0           | 0           | 0           | 0           | —                | —                | 0                     | 0                     | 0                     | 0                     | 1                     | 1                     |
| 87              | 00EE            | GPIO87 | CNTXC               | PCSD3             |                      |             | —                                                                      | 0           | 0           | 0           | 0           | —                | —                | 0                     | 0                     | 0                     | 0                     | 1                     | 1                     |

Table continues on the next page...



**Table 8-3. SIU\_PCRn settings  
(continued)**

| P<br>C<br>R | Offset<br>(hex) | GPIO    | Primary<br>function | Alternate 1        | Alternate 2 | Alternate 3 | SIU_PCRn[3:15]<br>(SIU_PCRn[0:2] = reserved) |             |             |             |             |                  |                  |             |             |                  |                  |             |             |
|-------------|-----------------|---------|---------------------|--------------------|-------------|-------------|----------------------------------------------|-------------|-------------|-------------|-------------|------------------|------------------|-------------|-------------|------------------|------------------|-------------|-------------|
|             |                 |         |                     |                    |             |             | 3                                            | 4           | 5           | 6           | 7           | 8                | 9                | 10          | 11          | 12               | 13               | 14          | 15          |
|             |                 |         |                     |                    |             |             | P<br>A<br>2                                  | P<br>A<br>1 | P<br>A<br>0 | O<br>B<br>E | I<br>B<br>E | D<br>S<br>C<br>1 | D<br>S<br>C<br>0 | O<br>D<br>E | H<br>Y<br>S | S<br>R<br>C<br>1 | S<br>R<br>C<br>0 | W<br>P<br>E | W<br>P<br>S |
| 88          | 00F0            | GPIO88  | CNRXC               | PCSD4              | RXDE        |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | 1           |
| 89          | 00F2            | GPIO89  | TXDA <sup>2</sup>   |                    |             |             | —                                            | —           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | 1           |
| 90          | 00F4            | GPIO90  | RXDA <sup>2</sup>   |                    |             |             | —                                            | —           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | 1           |
| 91          | 00F6            | GPIO91  | TXDB <sup>2</sup>   | PCSD1              | MCANRX0     |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | 1           |
| 92          | 00F8            | GPIO92  | RXDB <sup>2</sup>   | PCSD5              |             |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | 1           |
| 93          | 00FA            | GPIO93  | SCKA                | PCSC1              | MCANTX0     | SENT1_B     | 0                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | 1           |
| 94          | 00FC            | GPIO94  | SINA                | PCSC2              |             |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | 1           |
| 95          | 00FE            | GPIO95  | SOUTA <sup>2</sup>  | PCSC5              |             |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | 1           |
| 96          | 0100            | GPIO96  | PCSA0               | PCSD2              |             |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | 1           |
| 97          | 0102            | GPIO97  | PCSA1 <sup>2</sup>  | PCSE0              |             |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | 1           |
| 98          | 0104            | GPIO98  | PCSA2               | SOUTE              | FEC_TXD2    | FEC_RXER    | 0                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | 1           |
| 99          | 0106            | GPIO99  | PCSA3               | SINE               | FEC_MDIO    | TXDD        | 0                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | 1           |
| 100         | 0108            | GPIO100 | PCSA4 <sup>2</sup>  | SCKE               | TXDE        | FEC_RXCLK   | 0                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | 1           |
| 101         | 010A            | GPIO101 | PCSA5               | ETRIG1             | RXDD        | FEC_TXD3    | 0                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | 1           |
| 102         | 010C            | GPIO102 | SCKB                |                    |             |             | —                                            | —           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | 1           |
| 103         | 010E            | GPIO103 | SINB                |                    |             |             | —                                            | —           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | 1           |
| 104         | 0110            | GPIO104 | SOUTB               |                    |             |             | —                                            | —           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | 1           |
| 105         | 0112            | GPIO105 | PCSB0 <sup>2</sup>  | PCSD2              |             |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | 1           |
| 106         | 0114            | GPIO106 | PCSB1 <sup>2</sup>  | PCSD0              |             |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | 1           |
| 107         | 0116            | GPIO107 | PCSB2               | SOUTC              | SENT2_B     | FEC_RXD3    | 0                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | 1           |
| 108         | 0118            | GPIO108 | PCSB3               | SINC <sup>2</sup>  | SENT3_B     |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | 1           |
| 109         | 011A            | GPIO109 | PCSB4               | SCKC <sup>2</sup>  | SENT4_B     | FEC_MDC     | 0                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | 1           |
| 110         | 011C            | GPIO110 | PCSB5               | PCSC0 <sup>2</sup> | SENT5_B     | FEC_RXD2    | 0                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | 1           |
| 113         | 0122            | GPIO113 | TCRCLKA             | IRQ7               |             |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | 1           |
| 114         | 0124            | GPIO114 | ETPUA0              | ETPUA12            |             |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |
| 115         | 0126            | GPIO115 | ETPUA1              | ETPUA13            |             |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |
| 116         | 0128            | GPIO116 | ETPUA2              | ETPUA14            | RCH0_A      |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |
| 117         | 012A            | GPIO117 | ETPUA3              | ETPUA15            | RCH1_A      | ERROR0      | 0                                            | 0           | 0           | 0           | 1           | —                | —                | 0           | 0           | 0                | 0                | 0           | U           |
| 118         | 012C            | GPIO118 | ETPUA4              | ETPUA16            | RCH2_A      |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |
| 119         | 012E            | GPIO119 | ETPUA5              | ETPUA17            | RCH3_A      |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |
| 120         | 0130            | GPIO120 | ETPUA6              | ETPUA18            | RCH4_A      |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |
| 121         | 0132            | GPIO121 | ETPUA7              | ETPUA19            | RCH5_A      |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |

Table continues on the next page...

**Table 8-3. SIU\_PCRn settings  
(continued)**

| P<br>C<br>R | Offset<br>(hex) | GPIO    | Primary<br>function  | Alternate 1 | Alternate 2          | Alternate 3 | SIU_PCRn[3:15]<br>(SIU_PCRn[0:2] = reserved) |             |             |             |             |                  |                  |             |             |                  |                  |             |             |  |
|-------------|-----------------|---------|----------------------|-------------|----------------------|-------------|----------------------------------------------|-------------|-------------|-------------|-------------|------------------|------------------|-------------|-------------|------------------|------------------|-------------|-------------|--|
|             |                 |         |                      |             |                      |             | 3                                            | 4           | 5           | 6           | 7           | 8                | 9                | 10          | 11          | 12               | 13               | 14          | 15          |  |
|             |                 |         |                      |             |                      |             | P<br>A<br>2                                  | P<br>A<br>1 | P<br>A<br>0 | O<br>B<br>E | I<br>B<br>E | D<br>S<br>C<br>1 | D<br>S<br>C<br>0 | O<br>D<br>E | H<br>Y<br>S | S<br>R<br>C<br>1 | S<br>R<br>C<br>0 | W<br>P<br>E | W<br>P<br>S |  |
| 122         | 0134            | GPIO122 | ETPUA8               | ETPUA20     | RCH6_A               |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 123         | 0136            | GPIO123 | ETPUA9               | ETPUA21     | RCH7_A               |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 124         | 0138            | GPIO124 | ETPUA10              | ETPUA22     | RCH8_A               | ERRORIN     | 0                                            | 0           | 0           | 0           | 1           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 125         | 013A            | GPIO125 | ETPUA11              | ETPUA23     | RCH9_A               | ERROR1      | 0                                            | 0           | 0           | 0           | 1           | —                | —                | 0           | 0           | 0                | 0                | 0           | U           |  |
| 126         | 013C            | GPIO126 | ETPUA12              | PCSB1       |                      |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 127         | 013E            | GPIO127 | ETPUA13              | PCSB3       |                      |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 128         | 0140            | GPIO128 | ETPUA14              | PCSB4       |                      |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 129         | 0142            | GPIO129 | ETPUA15              | PCSB5       | MCANTX0              |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 130         | 0144            | GPIO130 | ETPUA16              | PCSD1       |                      |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 131         | 0146            | GPIO131 | ETPUA17              | PCSD2       |                      |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 132         | 0148            | GPIO132 | ETPUA18              | PCSD3       |                      |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 133         | 014A            | GPIO133 | ETPUA19              | PCSD4       | MCANRX0 <sup>2</sup> |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 134         | 014C            | GPIO134 | ETPUA20              | IRQ8        |                      |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 135         | 014E            | GPIO135 | ETPUA21              | IRQ9        |                      |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 136         | 0150            | GPIO136 | ETPUA22              | IRQ10       | RCH0_A               |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 137         | 0152            | GPIO137 | ETPUA23              | IRQ11       | RCH1_A               |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 138         | 0154            | GPIO138 | ETPUA24              | IRQ12       | RCH2_A               |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 139         | 0156            | GPIO139 | ETPUA25              | IRQ13       | RCH3_A               |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 140         | 0158            | GPIO140 | ETPUA26              | IRQ14       | RCH4_A               |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 141         | 015A            | GPIO141 | ETPUA27              | IRQ15       | RCH5_A               |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 142         | 015C            | GPIO142 | ETPUA28              | PCSC1       | RCH6_A               |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 143         | 015E            | GPIO143 | ETPUA29              | PCSC2       | RCH7_A               |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 144         | 0160            | GPIO144 | ETPUA30              | PCSC3       | RCH8_A               |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 145         | 0162            | GPIO145 | ETPUA31              | PCSC4       | RCH9_A               |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 146         | 0164            | GPIO146 | TCRCLKB <sup>2</sup> | IRQ6        |                      |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | 1           |  |
| 147         | 0166            | GPIO147 | ETPUB0               | ETPUB16     |                      |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 148         | 0168            | GPIO148 | ETPUB1               | ETPUB17     |                      |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 149         | 016A            | GPIO149 | ETPUB2               | ETPUB18     |                      |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 150         | 016C            | GPIO150 | ETPUB3               | ETPUB19     |                      |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 151         | 016E            | GPIO151 | ETPUB4               | ETPUB20     |                      |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 152         | 0170            | GPIO152 | ETPUB5               | ETPUB21     |                      |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 153         | 0172            | GPIO153 | ETPUB6               | ETPUB22     |                      |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |

Table continues on the next page...

**Table 8-3. SIU\_PCR<sub>n</sub> settings  
(continued)**

| P<br>C<br>R | Offset<br>(hex) | GPIO    | Primary<br>function | Alternate 1 | Alternate 2 | Alternate 3 | SIU_PCRn[3:15]<br>(SIU_PCRn[0:2] = reserved) |             |             |             |             |                  |                  |             |             |                  |                  |             |             |  |
|-------------|-----------------|---------|---------------------|-------------|-------------|-------------|----------------------------------------------|-------------|-------------|-------------|-------------|------------------|------------------|-------------|-------------|------------------|------------------|-------------|-------------|--|
|             |                 |         |                     |             |             |             | 3                                            | 4           | 5           | 6           | 7           | 8                | 9                | 10          | 11          | 12               | 13               | 14          | 15          |  |
|             |                 |         |                     |             |             |             | P<br>A<br>2                                  | P<br>A<br>1 | P<br>A<br>0 | O<br>B<br>E | I<br>B<br>E | D<br>S<br>C<br>1 | D<br>S<br>C<br>0 | O<br>D<br>E | H<br>Y<br>S | S<br>R<br>C<br>1 | S<br>R<br>C<br>0 | W<br>P<br>E | W<br>P<br>S |  |
| 154         | 0174            | GPIO154 | ETPUB7              | ETPUB23     |             |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 155         | 0176            | GPIO155 | ETPUB8              | ETPUB24     |             |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 156         | 0178            | GPIO156 | ETPUB9              | ETPUB25     |             |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 157         | 017A            | GPIO157 | ETPUB10             | ETPUB26     |             |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 158         | 017C            | GPIO158 | ETPUB11             | ETPUB27     |             |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 159         | 017E            | GPIO159 | ETPUB12             | ETPUB28     |             |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 160         | 0180            | GPIO160 | ETPUB13             | ETPUB29     |             |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 161         | 0182            | GPIO161 | ETPUB14             | ETPUB30     |             |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 162         | 0184            | GPIO162 | ETPUB15             | ETPUB31     |             |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 163         | 0186            | GPIO163 | ETPUB16             | PCSA1       |             |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 164         | 0188            | GPIO164 | ETPUB17             | PCSA2       |             |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 165         | 018A            | GPIO165 | ETPUB18             | PCSA3       |             |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 166         | 018C            | GPIO166 | ETPUB19             | PCSA4       |             |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 167         | 018E            | GPIO167 | ETPUB20             |             |             |             | —                                            | —           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 168         | 0190            | GPIO168 | ETPUB21             |             |             |             | —                                            | —           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 169         | 0192            | GPIO169 | ETPUB22             | RCH0_B      |             |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 170         | 0194            | GPIO170 | ETPUB23             | RCH1_B      |             |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 171         | 0196            | GPIO171 | ETPUB24             | RCH2_B      |             |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 172         | 0198            | GPIO172 | ETPUB25             | RCH3_B      |             |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 173         | 019A            | GPIO173 | ETPUB26             | RCH4_B      |             |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 174         | 019C            | GPIO174 | ETPUB27             | RCH5_B      |             |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 175         | 019E            | GPIO175 | ETPUB28             | RCH6_B      |             |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 176         | 01A0            | GPIO176 | ETPUB29             | RCH7_B      |             |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 177         | 01A2            | GPIO177 | ETPUB30             | RCH8_B      |             |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 178         | 01A4            | GPIO178 | ETPUB31             | RCH9_B      |             |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 179         | 01A6            | GPIO179 | EMIOS0              | ETPUA0      | RCH0_C      |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 180         | 01A8            | GPIO180 | EMIOS1              | ETPUA1      | RCH1_C      | PCSE2       | 0                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 181         | 01AA            | GPIO181 | EMIOS2              | ETPUA2      | RCH2_C      | PCSE3       | 0                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 182         | 01AC            | GPIO182 | EMIOS3              | ETPUA3      | RCH3_C      | PCSE4       | 0                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 183         | 01AE            | GPIO183 | EMIOS4              | ETPUA4      | RCH4_C      | PCSE5       | 0                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 184         | 01B0            | GPIO184 | EMIOS5              | ETPUA5      | RCH5_C      |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 185         | 01B2            | GPIO185 | EMIOS6              | ETPUA6      | RCH6_C      |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |

Table continues on the next page...

**Table 8-3. SIU\_PCRn settings  
(continued)**

| P<br>C<br>R | Offset<br>(hex) | GPIO     | Primary<br>function | Alternate 1        | Alternate 2 | Alternate 3 | SIU_PCRn[3:15]<br>(SIU_PCRn[0:2] = reserved) |                                      |                                      |                                      |                                      |                                      |                                      |                                      |                                      |                                      |                                      |                                      |             |  |
|-------------|-----------------|----------|---------------------|--------------------|-------------|-------------|----------------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|-------------|--|
|             |                 |          |                     |                    |             |             | 3                                            | 4                                    | 5                                    | 6                                    | 7                                    | 8                                    | 9                                    | 10                                   | 11                                   | 12                                   | 13                                   | 14                                   | 15          |  |
|             |                 |          |                     |                    |             |             | P<br>A<br>2                                  | P<br>A<br>1                          | P<br>A<br>0                          | O<br>B<br>E                          | I<br>B<br>E                          | D<br>S<br>C<br>1                     | D<br>S<br>C<br>0                     | O<br>D<br>E                          | H<br>Y<br>S                          | S<br>R<br>C<br>1                     | S<br>R<br>C<br>0                     | W<br>P<br>E                          | W<br>P<br>S |  |
| 186         | 01B4            | GPIO186  | EMIOS7              | ETPUA7             | RCH7_C      |             | —                                            | 0                                    | 0                                    | 0                                    | 0                                    | —                                    | —                                    | 0                                    | 0                                    | 0                                    | 0                                    | 1                                    | U           |  |
| 187         | 01B6            | GPIO187  | EMIOS8              | ETPUA8             | RCH8_C      |             | —                                            | 0                                    | 0                                    | 0                                    | 0                                    | —                                    | —                                    | 0                                    | 0                                    | 0                                    | 0                                    | 1                                    | U           |  |
| 188         | 01B8            | GPIO188  | EMIOS9              | ETPUA9             | RCH9_C      |             | —                                            | 0                                    | 0                                    | 0                                    | 0                                    | —                                    | —                                    | 0                                    | 0                                    | 0                                    | 0                                    | 1                                    | U           |  |
| 189         | 01BA            | GPIO189  | EMIOS10             | SCKD               |             |             | —                                            | 0                                    | 0                                    | 0                                    | 0                                    | —                                    | —                                    | 0                                    | 0                                    | 0                                    | 0                                    | 1                                    | U           |  |
| 190         | 01BC            | GPIO190  | EMIOS11             | SIND               |             |             | —                                            | 0                                    | 0                                    | 0                                    | 0                                    | —                                    | —                                    | 0                                    | 0                                    | 0                                    | 0                                    | 1                                    | U           |  |
| 191         | 01BE            | GPIO191  | EMIOS12             | SOUTC              |             |             | —                                            | 0                                    | 0                                    | 0                                    | 0                                    | —                                    | —                                    | 0                                    | 0                                    | 0                                    | 0                                    | 1                                    | U           |  |
| 192         | 01C0            | GPIO192  | EMIOS13             | SOUTD              |             |             | —                                            | 0                                    | 0                                    | 0                                    | 0                                    | —                                    | —                                    | 0                                    | 0                                    | 0                                    | 0                                    | 1                                    | U           |  |
| 193         | 01C2            | GPIO193  | EMIOS14             | IRQ0               | CNTXD       | RCH0_C      | 0                                            | 0                                    | 0                                    | 0                                    | 0                                    | —                                    | —                                    | 0                                    | 0                                    | 0                                    | 0                                    | 1                                    | U           |  |
| 194         | 01C4            | GPIO194  | EMIOS15             | IRQ1               | CNRXD       | RCH1_C      | 0                                            | 0                                    | 0                                    | 0                                    | 0                                    | —                                    | —                                    | 0                                    | 0                                    | 0                                    | 0                                    | 1                                    | U           |  |
| 195         | 01C6            | GPIO195  | EMIOS16             | ETPUB0             | MCANTX1     | RCH2_C      | 0                                            | 0                                    | 0                                    | 0                                    | 0                                    | —                                    | —                                    | 0                                    | 0                                    | 0                                    | 0                                    | 1                                    | U           |  |
| 196         | 01C8            | GPIO196  | EMIOS17             | ETPUB1             | MCANRX1     | RCH3_C      | 0                                            | 0                                    | 0                                    | 0                                    | 0                                    | —                                    | —                                    | 0                                    | 0                                    | 0                                    | 0                                    | 1                                    | U           |  |
| 197         | 01CA            | GPIO197  | EMIOS18             | ETPUB2             | RCH4_C      |             | —                                            | 0                                    | 0                                    | 0                                    | 0                                    | —                                    | —                                    | 0                                    | 0                                    | 0                                    | 0                                    | 1                                    | U           |  |
| 198         | 01CC            | GPIO198  | EMIOS19             | ETPUB3             | RCH5_C      |             | —                                            | 0                                    | 0                                    | 0                                    | 0                                    | —                                    | —                                    | 0                                    | 0                                    | 0                                    | 0                                    | 1                                    | U           |  |
| 199         | 01CE            | GPIO199  | EMIOS20             | ETPUB4             | RCH6_C      |             | —                                            | 0                                    | 0                                    | 0                                    | 0                                    | —                                    | —                                    | 0                                    | 0                                    | 0                                    | 0                                    | 1                                    | U           |  |
| 200         | 01D0            | GPIO200  | EMIOS21             | ETPUB5             | RCH7_C      |             | —                                            | 0                                    | 0                                    | 0                                    | 0                                    | —                                    | —                                    | 0                                    | 0                                    | 0                                    | 0                                    | 1                                    | U           |  |
| 201         | 01D2            | GPIO201  | EMIOS22             | ETPUB6             | RCH8_C      |             | —                                            | 0                                    | 0                                    | 0                                    | 0                                    | —                                    | —                                    | 0                                    | 0                                    | 0                                    | 0                                    | 1                                    | U           |  |
| 202         | 01D4            | GPIO202  | EMIOS23             | ETPUB7             | RCH9_C      |             | —                                            | 0                                    | 0                                    | 0                                    | 0                                    | —                                    | —                                    | 0                                    | 0                                    | 0                                    | 0                                    | 1                                    | U           |  |
| 203         | 01D6            | GPIO203  | EMIOS24             | PCSB0 <sup>2</sup> |             |             | —                                            | 0                                    | 0                                    | 0                                    | 0                                    | —                                    | —                                    | 0                                    | 0                                    | 0                                    | 0                                    | 1                                    | U           |  |
| 204         | 01D8            | GPIO204  | EMIOS25             | PCSB1              |             |             | —                                            | 0                                    | 0                                    | 0                                    | 0                                    | —                                    | —                                    | 0                                    | 0                                    | 0                                    | 0                                    | 1                                    | U           |  |
| 208         | 01E0            | Reserved | Reserved            | Reserved           | Reserved    | Reserved    | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d         | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d |             |  |
| 209         | 01E2            | Reserved | Reserved            | Reserved           | Reserved    | Reserved    | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d         | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d |             |  |

Table continues on the next page...

Table 8-3. SIU\_PCRn settings  
(continued)

| P<br>C<br>R      | Offset<br>(hex) | GPIO     | Primary<br>function   | Alternate 1       | Alternate 2 | Alternate 3 | SIU_PCRn[3:15]<br>(SIU_PCRn[0:2] = reserved) |                                      |                                      |                                      |                                      |                                      |                                      |                                      |                                      |                                      |                                      |                                      |                                      |  |
|------------------|-----------------|----------|-----------------------|-------------------|-------------|-------------|----------------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|--|
|                  |                 |          |                       |                   |             |             | 3                                            | 4                                    | 5                                    | 6                                    | 7                                    | 8                                    | 9                                    | 10                                   | 11                                   | 12                                   | 13                                   | 14                                   | 15                                   |  |
|                  |                 |          |                       |                   |             |             | P<br>A<br>2                                  | P<br>A<br>1                          | P<br>A<br>0                          | O<br>B<br>E                          | I<br>B<br>E                          | D<br>S<br>C<br>1                     | D<br>S<br>C<br>0                     | O<br>D<br>E<br>S                     | H<br>Y<br>S<br>C<br>1                | S<br>R<br>C<br>0                     | S<br>R<br>C<br>0                     | W<br>P<br>E                          | W<br>P<br>S                          |  |
| 210              | 01E4            | Reserved | Reserved              | Reserved          | Reserved    | Reserved    | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d         | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d |  |
| 211              | 01E6            | GPIO211  | BOOTCFG0 <sup>2</sup> | IRQ2              |             |             | —                                            | 0                                    | 1                                    | 0                                    | 0                                    | —                                    | —                                    | 0                                    | 1                                    | 0                                    | 0                                    | 1                                    | 0                                    |  |
| 212              | 01E8            | Reserved | Reserved              | Reserved          | Reserved    | Reserved    | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d         | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d |  |
| 213              | 01EA            | Reserved | Reserved              | Reserved          | Reserved    | Reserved    | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d         | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d |  |
| 214              | 01EC            |          | ENGCLK                |                   |             |             | —                                            | —                                    | —                                    | 1                                    | —                                    | —                                    | —                                    | —                                    | —                                    | 1                                    | 1                                    | —                                    | —                                    |  |
| 219              | 01F6            |          | MCKO                  |                   |             |             | —                                            | —                                    | —                                    | —                                    | —                                    | —                                    | —                                    | —                                    | —                                    | 1                                    | 1                                    | —                                    | —                                    |  |
| 220 <sup>1</sup> | 01F8            | GPIO220  | MDO0                  |                   |             |             | —                                            | —                                    | —                                    | 0                                    | 0                                    | —                                    | —                                    | 0                                    | 0                                    | 1                                    | 1                                    | 1                                    | 1                                    |  |
| 221 <sup>1</sup> | 01FA            | GPIO221  | MDO1                  |                   |             |             | —                                            | —                                    | —                                    | 0                                    | 0                                    | —                                    | —                                    | 0                                    | 0                                    | 1                                    | 1                                    | 1                                    | 0                                    |  |
| 222 <sup>1</sup> | 01FC            | GPIO222  | MDO2                  |                   |             |             | —                                            | —                                    | —                                    | 0                                    | 0                                    | —                                    | —                                    | 0                                    | 0                                    | 1                                    | 1                                    | 1                                    | 0                                    |  |
| 223 <sup>1</sup> | 01FE            | GPIO223  | MDO3                  | TXDE <sup>2</sup> |             |             | —                                            | 0                                    | 0                                    | 0                                    | 0                                    | —                                    | —                                    | 0                                    | 0                                    | 1                                    | 1                                    | 1                                    | 0                                    |  |
| 224              | 0200            |          | MSEO0                 |                   |             |             | —                                            | —                                    | —                                    | —                                    | —                                    | —                                    | —                                    | —                                    | —                                    | 1                                    | 1                                    | —                                    | —                                    |  |
| 225              | 0202            |          | MSEO1                 |                   |             |             | —                                            | —                                    | —                                    | —                                    | —                                    | —                                    | —                                    | —                                    | —                                    | 1                                    | 1                                    | —                                    | —                                    |  |
| 226              | 0204            |          | RDY                   |                   |             |             | —                                            | —                                    | —                                    | —                                    | —                                    | —                                    | —                                    | —                                    | —                                    | 1                                    | 1                                    | —                                    | —                                    |  |
| 227              | 0206            |          | EVTO                  |                   |             |             | —                                            | —                                    | —                                    | —                                    | —                                    | —                                    | —                                    | —                                    | —                                    | 1                                    | 1                                    | —                                    | —                                    |  |
| 228              | 0208            |          | TDO                   |                   |             |             | —                                            | —                                    | —                                    | —                                    | —                                    | —                                    | —                                    | —                                    | —                                    | 1                                    | 1                                    | —                                    | —                                    |  |
| 229              | 020A            |          | D_CLKOUT              |                   |             |             | —                                            | —                                    | —                                    | 1                                    | —                                    | 1                                    | 1                                    | —                                    | —                                    | —                                    | —                                    | —                                    | —                                    |  |
| 230              | 020C            |          | RSTOUT                |                   |             |             | —                                            | —                                    | —                                    | —                                    | —                                    | —                                    | —                                    | —                                    | —                                    | 1                                    | 1                                    | —                                    | —                                    |  |
| 231 <sup>3</sup> | 020E            | GPIO231  | MDO12                 |                   |             |             | —                                            | —                                    | —                                    | 0                                    | 0                                    | —                                    | —                                    | 0                                    | 0                                    | 1                                    | 1                                    | 1                                    | 0                                    |  |

Table continues on the next page...

**Table 8-3. SIU\_PCRn settings  
(continued)**

| P<br>C<br>R      | Offset<br>(hex) | GPIO    | Primary<br>function    | Alternate 1      | Alternate 2      | Alternate 3 | SIU_PCRn[3:15]<br>(SIU_PCRn[0:2] = reserved) |             |             |             |             |                  |                  |                  |                       |                       |                       |                  |                  |  |
|------------------|-----------------|---------|------------------------|------------------|------------------|-------------|----------------------------------------------|-------------|-------------|-------------|-------------|------------------|------------------|------------------|-----------------------|-----------------------|-----------------------|------------------|------------------|--|
|                  |                 |         |                        |                  |                  |             | 3                                            | 4           | 5           | 6           | 7           | 8                | 9                | 10               | 11                    | 12                    | 13                    | 14               | 15               |  |
|                  |                 |         |                        |                  |                  |             | P<br>A<br>2                                  | P<br>A<br>1 | P<br>A<br>0 | O<br>B<br>E | I<br>B<br>E | D<br>S<br>C<br>1 | D<br>S<br>C<br>0 | O<br>D<br>E<br>S | H<br>Y<br>S<br>C<br>1 | S<br>R<br>R<br>C<br>0 | S<br>R<br>R<br>C<br>1 | W<br>P<br>E<br>S | W<br>P<br>E<br>S |  |
| 232 <sup>1</sup> | 0210            | GPIO232 | MDO13                  |                  |                  |             | —                                            | —           | —           | 0           | 0           | —                | —                | 0                | 0                     | 1                     | 1                     | 1                | 0                |  |
| 233 <sup>1</sup> | 0212            | GPIO233 | MDO14                  |                  |                  |             | —                                            | —           | —           | 0           | 0           | —                | —                | 0                | 0                     | 1                     | 1                     | 1                | 0                |  |
| 234 <sup>1</sup> | 0214            | GPIO234 | MDO15                  |                  |                  |             | —                                            | —           | —           | 0           | 0           | —                | —                | 0                | 0                     | 1                     | 1                     | 1                | 0                |  |
| 235              | 0216            | GPIO235 | SCKC <sup>2</sup>      | SCK_C_LVD<br>S+  |                  |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0                | 0                     | 0                     | 0                     | 1                | 1                |  |
| 236              | 0218            | GPIO236 | SINC <sup>2</sup>      | SCK_C_LVD<br>S-  |                  |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0                | 0                     | 0                     | 0                     | 1                | 1                |  |
| 237              | 021A            | GPIO237 | SOUTC                  | SOUT_C_LV<br>DS+ |                  |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0                | 0                     | 0                     | 0                     | 1                | 1                |  |
| 238              | 021C            | GPIO238 | PCSC0 <sup>2</sup>     | SOUT_C_LV<br>DS- |                  |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0                | 0                     | 0                     | 0                     | 1                | 1                |  |
| 239              | 021E            | GPIO239 | PCSC1                  | TXDD             |                  |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0                | 0                     | 0                     | 0                     | 1                | 1                |  |
| 240              | 0220            | GPIO240 | PCSC2                  | SCK_D_LVD<br>S+  | SCKD             |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0                | 0                     | 0                     | 0                     | 1                | 1                |  |
| 241              | 0222            | GPIO241 | PCSC3                  | SCK_D_LVD<br>S-  | SIND             |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0                | 0                     | 0                     | 0                     | 1                | 1                |  |
| 242              | 0224            | GPIO242 | PCSC4                  | MCANTX0          | SOUT_D_LV<br>DS+ | SOUTD       | 0                                            | 0           | 0           | 0           | 0           | —                | —                | 0                | 0                     | 0                     | 0                     | 1                | 1                |  |
| 243              | 0226            | GPIO243 | PCSC5                  | MCANRX0          | SOUT_D_LV<br>DS- | PCSD0       | 0                                            | 0           | 0           | 0           | 0           | —                | —                | 0                | 0                     | 0                     | 0                     | 1                | 1                |  |
| 244              | 0228            | GPIO244 | TXDC                   | ETRIG0           |                  |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0                | 0                     | 0                     | 0                     | 1                | 1                |  |
| 245              | 022A            | GPIO245 | RXDC                   |                  |                  |             | —                                            | —           | 0           | 0           | 0           | —                | —                | 0                | 0                     | 0                     | 0                     | 1                | 1                |  |
| 246              | 022C            | GPIO246 | CNTXD                  |                  | MCANTX1          |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0                | 0                     | 0                     | 0                     | 1                | 1                |  |
| 247              | 022E            | GPIO247 | CNRXD <sup>2</sup>     |                  | MCANRX1          |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0                | 0                     | 0                     | 0                     | 1                | 1                |  |
| 248              | 0230            | GPIO248 | FEC_TXD0 <sup>2</sup>  | SENT0_A          |                  |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0                | 0                     | 1                     | 1                     | 1                | 1                |  |
| 249              | 0232            | GPIO249 | FEC_RX_DV <sup>2</sup> | SENT1_A          |                  |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0                | 0                     | 1                     | 1                     | 1                | 1                |  |
| 250              | 0234            | GPIO250 | FEC_RXD0 <sup>2</sup>  | SENT2_A          |                  |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0                | 0                     | 1                     | 1                     | 1                | 1                |  |
| 251              | 0236            | GPIO251 | FEC_TXD1 <sup>2</sup>  | SENT3_A          |                  |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0                | 0                     | 1                     | 1                     | 1                | 1                |  |
| 252              | 0238            | GPIO252 | FEC_TX_EN <sup>2</sup> | SENT4_A          |                  |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0                | 0                     | 1                     | 1                     | 1                | 1                |  |
| 253              | 023A            | GPIO253 | FEC_RXD1 <sup>2</sup>  | SENT5_A          |                  |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0                | 0                     | 1                     | 1                     | 1                | 1                |  |
| 256              | 0240            | GPIO256 | D_CS0                  |                  |                  |             | —                                            | —           | 0           | 0           | 0           | 1                | 1                | 0                | 0                     | —                     | —                     | 1                | 1                |  |
| 257              | 0242            | GPIO257 | D_CS2                  | D_ADD_DAT<br>31  |                  |             | —                                            | 0           | 0           | 0           | 0           | 1                | 1                | 0                | 0                     | —                     | —                     | 1                | 1                |  |
| 258              | 0244            | GPIO258 | D_CS3                  | D_TEA            |                  |             | —                                            | 0           | 0           | 0           | 0           | 1                | 1                | 0                | 0                     | —                     | —                     | 1                | 1                |  |

Table continues on the next page...

**Table 8-3. SIU\_PCR<sub>n</sub> settings  
(continued)**

| P<br>C<br>R | Offset<br>(hex) | GPIO    | Primary<br>function | Alternate 1     | Alternate 2 | Alternate 3 | SIU_PCRn[3:15]<br>(SIU_PCRn[0:2] = reserved) |             |             |             |             |                  |                  |             |             |                  |                  |             |             |  |
|-------------|-----------------|---------|---------------------|-----------------|-------------|-------------|----------------------------------------------|-------------|-------------|-------------|-------------|------------------|------------------|-------------|-------------|------------------|------------------|-------------|-------------|--|
|             |                 |         |                     |                 |             |             | 3                                            | 4           | 5           | 6           | 7           | 8                | 9                | 10          | 11          | 12               | 13               | 14          | 15          |  |
|             |                 |         |                     |                 |             |             | P<br>A<br>2                                  | P<br>A<br>1 | P<br>A<br>0 | O<br>B<br>E | I<br>B<br>E | D<br>S<br>C<br>1 | D<br>S<br>C<br>0 | O<br>D<br>E | H<br>Y<br>S | S<br>R<br>C<br>1 | S<br>R<br>C<br>0 | W<br>P<br>E | W<br>P<br>S |  |
| 259         | 0246            | GPIO259 | D_ADD12             |                 |             |             | —                                            | —           | 0           | 0           | 0           | 1                | 1                | 0           | 0           | —                | —                | 1           | 1           |  |
| 260         | 0248            | GPIO260 | D_ADD13             |                 |             |             | —                                            | —           | 0           | 0           | 0           | 1                | 1                | 0           | 0           | —                | —                | 1           | 1           |  |
| 261         | 024A            | GPIO261 | D_ADD14             |                 |             |             | —                                            | —           | 0           | 0           | 0           | 1                | 1                | 0           | 0           | —                | —                | 1           | 1           |  |
| 262         | 024C            | GPIO262 | D_ADD15             |                 |             |             | —                                            | —           | 0           | 0           | 0           | 1                | 1                | 0           | 0           | —                | —                | 1           | 1           |  |
| 263         | 024E            | GPIO263 | D_ADD16             | D_ADD_DAT<br>16 |             |             | —                                            | 0           | 0           | 0           | 0           | 1                | 1                | 0           | 0           | —                | —                | 1           | 1           |  |
| 264         | 0250            | GPIO264 | D_ADD17             | D_ADD_DAT<br>17 |             |             | —                                            | 0           | 0           | 0           | 0           | 1                | 1                | 0           | 0           | —                | —                | 1           | 1           |  |
| 265         | 0252            | GPIO265 | D_ADD18             | D_ADD_DAT<br>18 |             |             | —                                            | 0           | 0           | 0           | 0           | 1                | 1                | 0           | 0           | —                | —                | 1           | 1           |  |
| 266         | 0254            | GPIO266 | D_ADD19             | D_ADD_DAT<br>19 |             |             | —                                            | 0           | 0           | 0           | 0           | 1                | 1                | 0           | 0           | —                | —                | 1           | 1           |  |
| 267         | 0256            | GPIO267 | D_ADD20             | D_ADD_DAT<br>20 |             |             | —                                            | 0           | 0           | 0           | 0           | 1                | 1                | 0           | 0           | —                | —                | 1           | 1           |  |
| 268         | 0258            | GPIO268 | D_ADD21             | D_ADD_DAT<br>21 |             |             | —                                            | 0           | 0           | 0           | 0           | 1                | 1                | 0           | 0           | —                | —                | 1           | 1           |  |
| 269         | 025A            | GPIO269 | D_ADD22             | D_ADD_DAT<br>22 |             |             | —                                            | 0           | 0           | 0           | 0           | 1                | 1                | 0           | 0           | —                | —                | 1           | 1           |  |
| 270         | 025C            | GPIO270 | D_ADD23             | D_ADD_DAT<br>23 |             |             | —                                            | 0           | 0           | 0           | 0           | 1                | 1                | 0           | 0           | —                | —                | 1           | 1           |  |
| 271         | 025E            | GPIO271 | D_ADD24             | D_ADD_DAT<br>24 |             |             | —                                            | 0           | 0           | 0           | 0           | 1                | 1                | 0           | 0           | —                | —                | 1           | 1           |  |
| 272         | 0260            | GPIO272 | D_ADD25             | D_ADD_DAT<br>25 |             |             | —                                            | 0           | 0           | 0           | 0           | 1                | 1                | 0           | 0           | —                | —                | 1           | 1           |  |
| 273         | 0262            | GPIO273 | D_ADD26             | D_ADD_DAT<br>26 |             |             | —                                            | 0           | 0           | 0           | 0           | 1                | 1                | 0           | 0           | —                | —                | 1           | 1           |  |
| 274         | 0264            | GPIO274 | D_ADD27             | D_ADD_DAT<br>27 |             |             | —                                            | 0           | 0           | 0           | 0           | 1                | 1                | 0           | 0           | —                | —                | 1           | 1           |  |
| 275         | 0266            | GPIO275 | D_ADD28             | D_ADD_DAT<br>28 |             |             | —                                            | 0           | 0           | 0           | 0           | 1                | 1                | 0           | 0           | —                | —                | 1           | 1           |  |
| 276         | 0268            | GPIO276 | D_ADD29             | D_ADD_DAT<br>29 |             |             | —                                            | 0           | 0           | 0           | 0           | 1                | 1                | 0           | 0           | —                | —                | 1           | 1           |  |
| 277         | 026A            | GPIO277 | D_ADD30             | D_ADD_DAT<br>30 |             |             | —                                            | 0           | 0           | 0           | 0           | 1                | 1                | 0           | 0           | —                | —                | 1           | 1           |  |
| 278         | 026C            | GPIO278 | D_DAT0              |                 |             |             | —                                            | —           | 0           | 0           | 0           | —                | —                | 0           | 0           | 1                | 1                | 1           | 1           |  |
| 279         | 026E            | GPIO279 | D_DAT1              |                 |             |             | —                                            | —           | 0           | 0           | 0           | —                | —                | 0           | 0           | 1                | 1                | 1           | 1           |  |
| 280         | 0270            | GPIO280 | D_DAT2              |                 |             |             | —                                            | —           | 0           | 0           | 0           | —                | —                | 0           | 0           | 1                | 1                | 1           | 1           |  |

Table continues on the next page...

**Table 8-3. SIU\_PCRn settings  
(continued)**

| P<br>C<br>R | Offset<br>(hex) | GPIO    | Primary<br>function | Alternate 1        | Alternate 2 | Alternate 3 | SIU_PCRn[3:15]<br>(SIU_PCRn[0:2] = reserved) |             |             |             |             |                  |                  |                  |                       |                       |                       |             |             |  |
|-------------|-----------------|---------|---------------------|--------------------|-------------|-------------|----------------------------------------------|-------------|-------------|-------------|-------------|------------------|------------------|------------------|-----------------------|-----------------------|-----------------------|-------------|-------------|--|
|             |                 |         |                     |                    |             |             | 3                                            | 4           | 5           | 6           | 7           | 8                | 9                | 10               | 11                    | 12                    | 13                    | 14          | 15          |  |
|             |                 |         |                     |                    |             |             | P<br>A<br>2                                  | P<br>A<br>1 | P<br>A<br>0 | O<br>B<br>E | I<br>B<br>E | D<br>S<br>C<br>1 | D<br>S<br>C<br>0 | O<br>D<br>E<br>S | H<br>Y<br>S<br>C<br>1 | S<br>R<br>R<br>C<br>0 | S<br>R<br>R<br>C<br>1 | W<br>P<br>E | W<br>P<br>S |  |
| 281         | 0272            | GPIO281 | D_DAT3              |                    |             |             | —                                            | —           | 0           | 0           | 0           | —                | —                | 0                | 0                     | 1                     | 1                     | 1           | 1           |  |
| 282         | 0274            | GPIO282 | D_DAT4              |                    |             |             | —                                            | —           | 0           | 0           | 0           | —                | —                | 0                | 0                     | 1                     | 1                     | 1           | 1           |  |
| 283         | 0276            | GPIO283 | D_DAT5              |                    |             |             | —                                            | —           | 0           | 0           | 0           | —                | —                | 0                | 0                     | 1                     | 1                     | 1           | 1           |  |
| 284         | 0278            | GPIO284 | D_DAT6              |                    |             |             | —                                            | —           | 0           | 0           | 0           | —                | —                | 0                | 0                     | 1                     | 1                     | 1           | 1           |  |
| 285         | 027A            | GPIO285 | D_DAT7              |                    |             |             | —                                            | —           | 0           | 0           | 0           | —                | —                | 0                | 0                     | 1                     | 1                     | 1           | 1           |  |
| 286         | 027C            | GPIO286 | D_DAT8              |                    |             |             | —                                            | —           | 0           | 0           | 0           | —                | —                | 0                | 0                     | 1                     | 1                     | 1           | 1           |  |
| 287         | 027E            | GPIO287 | D_DAT9              |                    |             |             | —                                            | —           | 0           | 0           | 0           | —                | —                | 0                | 0                     | 1                     | 1                     | 1           | 1           |  |
| 288         | 0280            | GPIO288 | D_DAT10             |                    |             |             | —                                            | —           | 0           | 0           | 0           | —                | —                | 0                | 0                     | 1                     | 1                     | 1           | 1           |  |
| 289         | 0282            | GPIO289 | D_DAT11             |                    |             |             | —                                            | —           | 0           | 0           | 0           | —                | —                | 0                | 0                     | 1                     | 1                     | 1           | 1           |  |
| 290         | 0284            | GPIO290 | D_DAT12             |                    |             |             | —                                            | —           | 0           | 0           | 0           | —                | —                | 0                | 0                     | 1                     | 1                     | 1           | 1           |  |
| 291         | 0286            | GPIO291 | D_DAT13             |                    |             |             | —                                            | —           | 0           | 0           | 0           | —                | —                | 0                | 0                     | 1                     | 1                     | 1           | 1           |  |
| 292         | 0288            | GPIO292 | D_DAT14             |                    |             |             | —                                            | —           | 0           | 0           | 0           | —                | —                | 0                | 0                     | 1                     | 1                     | 1           | 1           |  |
| 293         | 028A            | GPIO293 | D_DAT15             |                    |             |             | —                                            | —           | 0           | 0           | 0           | —                | —                | 0                | 0                     | 1                     | 1                     | 1           | 1           |  |
| 294         | 028C            | GPIO294 | D_RD_WR             |                    |             |             | —                                            | —           | 0           | 0           | 0           | 1                | 1                | 0                | 0                     | —                     | —                     | 1           | 1           |  |
| 295         | 028E            | GPIO295 | D_WE0               |                    |             |             | —                                            | —           | 0           | 0           | 0           | 1                | 1                | 0                | 0                     | —                     | —                     | 1           | 1           |  |
| 296         | 0290            | GPIO296 | D_WE1               |                    |             |             | —                                            | —           | 0           | 0           | 0           | 1                | 1                | 0                | 0                     | —                     | —                     | 1           | 1           |  |
| 297         | 0292            | GPIO297 | D_OE                |                    |             |             | —                                            | —           | 0           | 0           | 0           | 1                | 1                | 0                | 0                     | —                     | —                     | 1           | 1           |  |
| 298         | 0294            | GPIO298 | D_TS                |                    |             |             | —                                            | —           | 0           | 0           | 0           | 1                | 1                | 0                | 0                     | —                     | —                     | 1           | 1           |  |
| 299         | 0296            | GPIO299 | D_ALE               |                    |             |             | —                                            | —           | 0           | 0           | 0           | —                | —                | 0                | 0                     | 1                     | 1                     | 1           | 1           |  |
| 300         | 0298            | GPIO300 | D_TA                |                    |             |             | —                                            | —           | 0           | 0           | 0           | 1                | 1                | 0                | 0                     | —                     | —                     | 1           | 1           |  |
| 301         | 029A            | GPIO301 | D_CS1               |                    |             |             | —                                            | —           | 0           | 0           | 0           | 1                | 1                | 0                | 0                     | —                     | —                     | 1           | 1           |  |
| 302         | 029C            | GPIO302 | D_BDIP              |                    |             |             | —                                            | —           | 0           | 0           | 0           | 1                | 1                | 0                | 0                     | —                     | —                     | 1           | 1           |  |
| 303         | 029E            | GPIO303 | D_WE2               |                    |             |             | —                                            | —           | 0           | 0           | 0           | 1                | 1                | 0                | 0                     | —                     | —                     | 1           | 1           |  |
| 304         | 02A0            | GPIO304 | D_WE3               |                    |             |             | —                                            | —           | 0           | 0           | 0           | 1                | 1                | 0                | 0                     | —                     | —                     | 1           | 1           |  |
| 305         | 02A2            | GPIO305 | D_ADD9              |                    |             |             | —                                            | —           | 0           | 0           | 0           | 1                | 1                | 0                | 0                     | —                     | —                     | 1           | 1           |  |
| 306         | 02A4            | GPIO306 | D_ADD10             |                    |             |             | —                                            | —           | 0           | 0           | 0           | 1                | 1                | 0                | 0                     | —                     | —                     | 1           | 1           |  |
| 307         | 02A6            | GPIO307 | D_ADD11             |                    |             |             | —                                            | —           | 0           | 0           | 0           | 1                | 1                | 0                | 0                     | —                     | —                     | 1           | 1           |  |
| 432         | 03A0            | GPIO432 | EMIOS26             | PCSB2              |             |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0                | 0                     | 0                     | 0                     | 1           | U           |  |
| 433         | 03A2            | GPIO433 | EMIOS27             | PCSB3              |             |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0                | 0                     | 0                     | 0                     | 1           | U           |  |
| 434         | 03A4            | GPIO434 | EMIOS28             | PCSC0 <sup>2</sup> |             |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0                | 0                     | 0                     | 0                     | 1           | U           |  |
| 435         | 03A6            | GPIO435 | EMIOS29             | PCSC1              |             |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0                | 0                     | 0                     | 0                     | 1           | U           |  |
| 436         | 03A8            | GPIO436 | EMIOS30             | PCSC2              |             |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0                | 0                     | 0                     | 0                     | 1           | U           |  |

Table continues on the next page...



**Table 8-3. SIU\_PCRn settings  
(continued)**

| P<br>C<br>R | Offset<br>(hex) | GPIO     | Primary<br>function  | Alternate 1          | Alternate 2          | Alternate 3 | SIU_PCRn[3:15]<br>(SIU_PCRn[0:2] = reserved) |                                      |                                      |                                      |                                      |                                      |                                      |                                      |                                      |                                      |                                      |                                      |             |  |  |
|-------------|-----------------|----------|----------------------|----------------------|----------------------|-------------|----------------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|-------------|--|--|
|             |                 |          |                      |                      |                      |             | 3                                            | 4                                    | 5                                    | 6                                    | 7                                    | 8                                    | 9                                    | 10                                   | 11                                   | 12                                   | 13                                   | 14                                   | 15          |  |  |
|             |                 |          |                      |                      |                      |             | P<br>A<br>2                                  | P<br>A<br>1                          | P<br>A<br>0                          | O<br>B<br>E                          | I<br>B<br>E                          | D<br>S<br>C<br>1                     | D<br>S<br>C<br>0                     | O<br>D<br>E<br>S                     | H<br>Y<br>S<br>C<br>1                | S<br>R<br>P<br>C<br>0                | S<br>R<br>P<br>C<br>1                | W<br>P<br>E                          | W<br>P<br>S |  |  |
| 437         | 03AA            | GPIO437  | EMIOS31              | PCSC5                |                      |             | —                                            | 0                                    | 0                                    | 0                                    | 0                                    | —                                    | —                                    | 0                                    | 0                                    | 0                                    | 0                                    | 1                                    | U           |  |  |
| 440         | 03B0            | GPI440   | TCRCLKC              | IRQ6 <sup>2</sup>    | ANB40                |             | —                                            | 0                                    | 0                                    | —                                    | 0                                    | —                                    | —                                    | —                                    | 0                                    | —                                    | —                                    | 1                                    | 1           |  |  |
| 441         | 03B2            | GPI441   | ETPUC0               | SENT0_A              | ANB41                |             | —                                            | 0                                    | 0                                    | —                                    | 0                                    | —                                    | —                                    | —                                    | 0                                    | —                                    | —                                    | 1                                    | U           |  |  |
| 442         | 03B4            | GPI442   | ETPUC1               | SENT1_A              | ANB42                |             | —                                            | 0                                    | 0                                    | —                                    | 0                                    | —                                    | —                                    | —                                    | 0                                    | —                                    | —                                    | 1                                    | U           |  |  |
| 443         | 03B6            | GPIO443  | ETPUC2               | SENT3_A              |                      |             | —                                            | 0                                    | 0                                    | 0                                    | 0                                    | —                                    | —                                    | 0                                    | 0                                    | 0                                    | 0                                    | 1                                    | U           |  |  |
| 444         | 03B8            | GPI444   | ETPUC3               | SENT4_A              | ANB45                |             | —                                            | 0                                    | 0                                    | —                                    | 0                                    | —                                    | —                                    | —                                    | 0                                    | —                                    | —                                    | 1                                    | U           |  |  |
| 445         | 03BA            | GPIO445  | ETPUC4               | SENT5_A              | TCRCLKC              |             | —                                            | 0                                    | 0                                    | 0                                    | 0                                    | —                                    | —                                    | 0                                    | 0                                    | 0                                    | 0                                    | 1                                    | U           |  |  |
| 446         | 03BC            | GPIO446  | ETPUC5               | PCSE2                | SENT0_B              |             | —                                            | 0                                    | 0                                    | 0                                    | 0                                    | —                                    | —                                    | 0                                    | 0                                    | 0                                    | 0                                    | 1                                    | U           |  |  |
| 447         | 03BE            | GPIO447  | ETPUC6               | PCSE3                | SENT1_B <sup>2</sup> |             | —                                            | 0                                    | 0                                    | 0                                    | 0                                    | —                                    | —                                    | 0                                    | 0                                    | 0                                    | 0                                    | 1                                    | U           |  |  |
| 448         | 03C0            | GPIO448  | ETPUC7               | PCSE4                | SENT2_B <sup>2</sup> |             | —                                            | 0                                    | 0                                    | 0                                    | 0                                    | —                                    | —                                    | 0                                    | 0                                    | 0                                    | 0                                    | 1                                    | U           |  |  |
| 449         | 03C2            | GPIO449  | ETPUC8 <sup>2</sup>  | PCSE5                | SENT3_B <sup>2</sup> |             | —                                            | 0                                    | 0                                    | 0                                    | 0                                    | —                                    | —                                    | 0                                    | 0                                    | 0                                    | 0                                    | 1                                    | U           |  |  |
| 450         | 03C4            | GPI450   | ETPUC9 <sup>2</sup>  | IRQ0 <sup>2</sup>    | SENT4_B <sup>2</sup> | ANB44       | —                                            | 0                                    | 0                                    | —                                    | 0                                    | —                                    | —                                    | —                                    | 0                                    | —                                    | —                                    | 1                                    | U           |  |  |
| 451         | 03C6            | GPIO451  | ETPUC10 <sup>2</sup> | IRQ1 <sup>2</sup>    | SENT5_B <sup>2</sup> |             | —                                            | 0                                    | 0                                    | 0                                    | 0                                    | —                                    | —                                    | 0                                    | 0                                    | 0                                    | 0                                    | 1                                    | U           |  |  |
| 452         | 03C8            | GPIO452  | ETPUC11 <sup>2</sup> | IRQ2 <sup>2</sup>    |                      |             | —                                            | 0                                    | 0                                    | 0                                    | 0                                    | —                                    | —                                    | 0                                    | 0                                    | 0                                    | 0                                    | 1                                    | U           |  |  |
| 453         | 03CA            | GPIO453  | ETPUC12 <sup>2</sup> | IRQ3 <sup>2</sup>    | MCANTX1              |             | —                                            | 0                                    | 0                                    | 0                                    | 0                                    | —                                    | —                                    | 0                                    | 0                                    | 0                                    | 0                                    | 1                                    | U           |  |  |
| 454         | 03CC            | Reserved | Reserved             | Reserved             | Reserved             | Reserved    | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d         | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d |             |  |  |
| 455         | 03CE            | Reserved | Reserved             | Reserved             | Reserved             | Reserved    | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d         | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d |             |  |  |
| 456         | 03D0            | GPIO456  | ETPUC15 <sup>2</sup> | MCANRX1 <sup>2</sup> | RCH2_C               |             | —                                            | 0                                    | 0                                    | 0                                    | 0                                    | —                                    | —                                    | 0                                    | 0                                    | 0                                    | 0                                    | 1                                    | U           |  |  |
| 457         | 03D2            | GPIO457  | ETPUC16              | MCANTX1              | RCH3_C               |             | —                                            | 0                                    | 0                                    | 0                                    | 0                                    | —                                    | —                                    | 0                                    | 0                                    | 0                                    | 0                                    | 1                                    | U           |  |  |
| 458         | 03D4            | GPIO458  | ETPUC17              | TXDF                 | RCH4_C               |             | —                                            | 0                                    | 0                                    | 0                                    | 0                                    | —                                    | —                                    | 0                                    | 0                                    | 0                                    | 0                                    | 1                                    | U           |  |  |
| 459         | 03D6            | GPIO459  | ETPUC18              | RXDF                 |                      |             | —                                            | 0                                    | 0                                    | 0                                    | 0                                    | —                                    | —                                    | 0                                    | 0                                    | 0                                    | 0                                    | 1                                    | U           |  |  |
| 460         | 03D8            | GPIO460  | ETPUC19              | TXDA                 | SDOUT0_PSI<br>5B     |             | —                                            | 0                                    | 0                                    | 0                                    | 0                                    | —                                    | —                                    | 0                                    | 0                                    | 0                                    | 0                                    | 1                                    | U           |  |  |

Table continues on the next page...

**Table 8-3. SIU\_PCRn settings  
(continued)**

| P<br>C<br>R | Offset<br>(hex) | GPIO    | Primary<br>function  | Alternate 1          | Alternate 2     | Alternate 3 | SIU_PCRn[3:15]<br>(SIU_PCRn[0:2] = reserved) |             |             |             |             |                  |                  |             |             |                  |                  |             |             |  |
|-------------|-----------------|---------|----------------------|----------------------|-----------------|-------------|----------------------------------------------|-------------|-------------|-------------|-------------|------------------|------------------|-------------|-------------|------------------|------------------|-------------|-------------|--|
|             |                 |         |                      |                      |                 |             | 3                                            | 4           | 5           | 6           | 7           | 8                | 9                | 10          | 11          | 12               | 13               | 14          | 15          |  |
|             |                 |         |                      |                      |                 |             | P<br>A<br>2                                  | P<br>A<br>1 | P<br>A<br>0 | O<br>B<br>E | I<br>B<br>E | D<br>S<br>C<br>1 | D<br>S<br>C<br>0 | O<br>D<br>S | H<br>Y<br>S | S<br>R<br>C<br>1 | S<br>R<br>C<br>0 | W<br>P<br>E | W<br>P<br>S |  |
| 461         | 03DA            | GPIO461 | ETPUC20              | RXDA                 | SDINO_PSI5<br>B |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 462         | 03DC            | GPIO462 | ETPUC21              | TXDB <sup>2</sup>    |                 |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 463         | 03DE            | GPIO463 | ETPUC22              | RXDB <sup>2</sup>    |                 |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 464         | 03E0            | GPIO464 | ETPUC23              | PCSD5                | MAA0            | MAB0        | 0                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 465         | 03E2            | GPIO465 | ETPUC24              | PCSD4                | MAA1            | MAB1        | 0                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 466         | 03E4            | GPIO466 | ETPUC25              | PCSD3                | MAA2            | MAB2        | 0                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 467         | 03E6            | GPIO467 | ETPUC26              | PCSD2                | MAB0            |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 468         | 03E8            | GPIO468 | ETPUC27              | PCSD1                | RCH5_C          | MAB1        | 0                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 469         | 03EA            | GPIO469 | ETPUC28              | PCSD0 <sup>2</sup>   | RCH6_C          | MAB2        | 0                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 470         | 03EC            | GPIO470 | ETPUC29              | SCKD <sup>2</sup>    | RCH7_C          |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 471         | 03EE            | GPIO471 | ETPUC30              | SOUTD                | RCH8_C          |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 472         | 03F0            | GPIO472 | ETPUC31              | SIND <sup>2</sup>    | RCH9_C          |             | —                                            | 0           | 0           | 0           | 0           | —                | —                | 0           | 0           | 0                | 0                | 1           | U           |  |
| 474         | 03F4            | GPI474  | FEC_TXCLK<br>_REFCLK | SENT0_B <sup>2</sup> |                 |             | —                                            | 0           | 0           | —           | 0           | —                | —                | 0           | —           | —                | —                | 1           | 1           |  |
| 475         | 03F6            | GPI475  | SENT2_A <sup>2</sup> | ANB43                |                 |             | —                                            | 0           | 0           | —           | 0           | —                | —                | 0           | —           | —                | —                | 1           | U           |  |
| 476         | 03F8            | GPI476  | ANB8                 | ETPUA31              | EMIOS16         |             | —                                            | 0           | 0           | —           | 0           | —                | —                | 0           | —           | —                | —                | 1           | U           |  |
| 477         | 03FA            | GPI477  | ANB9                 | ETPUA30              | EMIOS17         |             | —                                            | 0           | 0           | —           | 0           | —                | —                | 0           | —           | —                | —                | 1           | U           |  |
| 478         | 03FC            | GPI478  | ANB10                | ETPUA29              | EMIOS18         |             | —                                            | 0           | 0           | —           | 0           | —                | —                | 0           | —           | —                | —                | 1           | U           |  |
| 479         | 03FE            | GPI479  | ANB11                | ETPUA28              | EMIOS19         |             | —                                            | 0           | 0           | —           | 0           | —                | —                | 0           | —           | —                | —                | 1           | U           |  |
| 480         | 0400            | GPI480  | ANB12                | ETPUA27              | EMIOS20         |             | —                                            | 0           | 0           | —           | 0           | —                | —                | 0           | —           | —                | —                | 1           | U           |  |
| 481         | 0402            | GPI481  | ANB13                | ETPUA26              | EMIOS21         |             | —                                            | 0           | 0           | —           | 0           | —                | —                | 0           | —           | —                | —                | 1           | U           |  |
| 482         | 0404            | GPI482  | ANB14                | ETPUA25              | EMIOS22         |             | —                                            | 0           | 0           | —           | 0           | —                | —                | 0           | —           | —                | —                | 1           | U           |  |
| 483         | 0406            | GPI483  | ANB15                | ETPUA24              | EMIOS23         |             | —                                            | 0           | 0           | —           | 0           | —                | —                | 0           | —           | —                | —                | 1           | U           |  |
| 484         | 0408            | GPI484  | ANB16                | ETPUC15 <sup>2</sup> | EMIOS24         |             | —                                            | 0           | 0           | —           | 0           | —                | —                | 0           | —           | —                | —                | 1           | U           |  |
| 485         | 040A            | GPI485  | ANB17                | ETPUC14 <sup>2</sup> | EMIOS25         |             | —                                            | 0           | 0           | —           | 0           | —                | —                | 0           | —           | —                | —                | 1           | U           |  |
| 486         | 040C            | GPI486  | ANB18                | ETPUC13 <sup>2</sup> | EMIOS26         |             | —                                            | 0           | 0           | —           | 0           | —                | —                | 0           | —           | —                | —                | 1           | U           |  |
| 487         | 040E            | GPI487  | ANB19                | ETPUC12 <sup>2</sup> | EMIOS27         |             | —                                            | 0           | 0           | —           | 0           | —                | —                | 0           | —           | —                | —                | 1           | U           |  |
| 488         | 0410            | GPI488  | ANB20                | ETPUC11 <sup>2</sup> | EMIOS28         |             | —                                            | 0           | 0           | —           | 0           | —                | —                | 0           | —           | —                | —                | 1           | U           |  |
| 489         | 0412            | GPI489  | ANB21                | ETPUC10 <sup>2</sup> | EMIOS29         |             | —                                            | 0           | 0           | —           | 0           | —                | —                | 0           | —           | —                | —                | 1           | U           |  |
| 490         | 0414            | GPI490  | ANB22                | ETPUC9 <sup>2</sup>  | EMIOS30         |             | —                                            | 0           | 0           | —           | 0           | —                | —                | 0           | —           | —                | —                | 1           | U           |  |
| 491         | 0416            | GPI491  | ANB23                | ETPUC8 <sup>2</sup>  | EMIOS31         |             | —                                            | 0           | 0           | —           | 0           | —                | —                | 0           | —           | —                | —                | 1           | U           |  |

1. The choice between MDO and GPIO functions is based on Nexus programming. If Nexus is enabled, this pin takes the role of MDO, otherwise it is GPIO. The OBE and IBE fields only apply when the pin is working as GPIO, otherwise it is an output-only MDO pin.
2. This input function is also present in other pins, but only one pin can be selected at any given time. Select the pin to drive the function using the SIU\_IMUX registers.
3. The choice between MDO and GPIO functions is based on Nexus programming. If Nexus is enabled and configured to use Full Port Mode, then this pin takes the role of MDO. If Nexus is not enabled or it is enabled but configured to use Reduced Port Mode, then this pin takes the role of GPIO. The OBE and IBE fields only apply when the pin is working as GPIO, otherwise it is an output-only MDO pin.

### SIU memory map

| Address offset (hex) | Register name                                      | Width (in bits) | Access | Reset value                 | Section/ page              |
|----------------------|----------------------------------------------------|-----------------|--------|-----------------------------|----------------------------|
| 4                    | MCU Identification Register (SIU_MIDR)             | 32              | R      | <a href="#">See section</a> | <a href="#">8.2.1/217</a>  |
| C                    | Reset Status Register (SIU_RSR)                    | 32              | w1c    | <a href="#">See section</a> | <a href="#">8.2.2/218</a>  |
| 10                   | System Reset Control Register (SIU_SRCR)           | 32              | R/W    | 0000_0000h                  | <a href="#">8.2.3/221</a>  |
| 14                   | External IRQ Status Register (SIU_EISR)            | 32              | R      | 0000_0000h                  | <a href="#">8.2.4/222</a>  |
| 18                   | DMA/Interrupt Request Enable Register (SIU_DIRER)  | 32              | R/W    | 0000_0000h                  | <a href="#">8.2.5/225</a>  |
| 1C                   | DMA/Interrupt Request Select Register (SIU_DIRSR)  | 32              | R/W    | 0000_0000h                  | <a href="#">8.2.6/228</a>  |
| 20                   | Overrun Status Register (SIU_OSR)                  | 32              | w1c    | 0000_0000h                  | <a href="#">8.2.7/230</a>  |
| 24                   | Overrun Request Enable Register (SIU_ORER)         | 32              | R/W    | 0000_0000h                  | <a href="#">8.2.8/232</a>  |
| 28                   | IRQ Rising-Edge Event Enable Register (SIU_IREER)  | 32              | R/W    | 0000_0000h                  | <a href="#">8.2.9/234</a>  |
| 2C                   | IRQ Falling-Edge Event Enable Register (SIU_IFEER) | 32              | R/W    | 0000_0000h                  | <a href="#">8.2.10/236</a> |
| 30                   | IRQ Digital Filter Register (SIU_IDFR)             | 32              | R/W    | 0000_0000h                  | <a href="#">8.2.11/239</a> |
| 34                   | IRQ Filtered Input Register (SIU_IFIR)             | 32              | R      | 0000_0000h                  | <a href="#">8.2.12/239</a> |
| D6                   | Pad Configuration Register (SIU_PCR75)             | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| D8                   | Pad Configuration Register (SIU_PCR76)             | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| DA                   | Pad Configuration Register (SIU_PCR77)             | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| DC                   | Pad Configuration Register (SIU_PCR78)             | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| DE                   | Pad Configuration Register (SIU_PCR79)             | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| E0                   | Pad Configuration Register (SIU_PCR80)             | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| E2                   | Pad Configuration Register (SIU_PCR81)             | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| E4                   | Pad Configuration Register (SIU_PCR82)             | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| E6                   | Pad Configuration Register (SIU_PCR83)             | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| E8                   | Pad Configuration Register (SIU_PCR84)             | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| EA                   | Pad Configuration Register (SIU_PCR85)             | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| EC                   | Pad Configuration Register (SIU_PCR86)             | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| EE                   | Pad Configuration Register (SIU_PCR87)             | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| F0                   | Pad Configuration Register (SIU_PCR88)             | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| F2                   | Pad Configuration Register (SIU_PCR89)             | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| F4                   | Pad Configuration Register (SIU_PCR90)             | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| F6                   | Pad Configuration Register (SIU_PCR91)             | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| F8                   | Pad Configuration Register (SIU_PCR92)             | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| FA                   | Pad Configuration Register (SIU_PCR93)             | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |

Table continues on the next page...

## SIU memory map (continued)

| Address offset (hex) | Register name                           | Width (in bits) | Access | Reset value                 | Section/ page              |
|----------------------|-----------------------------------------|-----------------|--------|-----------------------------|----------------------------|
| FC                   | Pad Configuration Register (SIU_PCR94)  | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| FE                   | Pad Configuration Register (SIU_PCR95)  | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 100                  | Pad Configuration Register (SIU_PCR96)  | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 102                  | Pad Configuration Register (SIU_PCR97)  | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 104                  | Pad Configuration Register (SIU_PCR98)  | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 106                  | Pad Configuration Register (SIU_PCR99)  | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 108                  | Pad Configuration Register (SIU_PCR100) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 10A                  | Pad Configuration Register (SIU_PCR101) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 10C                  | Pad Configuration Register (SIU_PCR102) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 10E                  | Pad Configuration Register (SIU_PCR103) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 110                  | Pad Configuration Register (SIU_PCR104) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 112                  | Pad Configuration Register (SIU_PCR105) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 114                  | Pad Configuration Register (SIU_PCR106) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 116                  | Pad Configuration Register (SIU_PCR107) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 118                  | Pad Configuration Register (SIU_PCR108) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 11A                  | Pad Configuration Register (SIU_PCR109) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 11C                  | Pad Configuration Register (SIU_PCR110) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 11E                  | Pad Configuration Register (SIU_PCR111) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 120                  | Pad Configuration Register (SIU_PCR112) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 122                  | Pad Configuration Register (SIU_PCR113) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 124                  | Pad Configuration Register (SIU_PCR114) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 126                  | Pad Configuration Register (SIU_PCR115) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 128                  | Pad Configuration Register (SIU_PCR116) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 12A                  | Pad Configuration Register (SIU_PCR117) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 12C                  | Pad Configuration Register (SIU_PCR118) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 12E                  | Pad Configuration Register (SIU_PCR119) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 130                  | Pad Configuration Register (SIU_PCR120) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 132                  | Pad Configuration Register (SIU_PCR121) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 134                  | Pad Configuration Register (SIU_PCR122) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 136                  | Pad Configuration Register (SIU_PCR123) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 138                  | Pad Configuration Register (SIU_PCR124) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 13A                  | Pad Configuration Register (SIU_PCR125) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 13C                  | Pad Configuration Register (SIU_PCR126) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 13E                  | Pad Configuration Register (SIU_PCR127) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 140                  | Pad Configuration Register (SIU_PCR128) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 142                  | Pad Configuration Register (SIU_PCR129) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 144                  | Pad Configuration Register (SIU_PCR130) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 146                  | Pad Configuration Register (SIU_PCR131) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 148                  | Pad Configuration Register (SIU_PCR132) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |

Table continues on the next page...

## SIU memory map (continued)

| Address offset (hex) | Register name                           | Width (in bits) | Access | Reset value                 | Section/ page              |
|----------------------|-----------------------------------------|-----------------|--------|-----------------------------|----------------------------|
| 14A                  | Pad Configuration Register (SIU_PCR133) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 14C                  | Pad Configuration Register (SIU_PCR134) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 14E                  | Pad Configuration Register (SIU_PCR135) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 150                  | Pad Configuration Register (SIU_PCR136) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 152                  | Pad Configuration Register (SIU_PCR137) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 154                  | Pad Configuration Register (SIU_PCR138) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 156                  | Pad Configuration Register (SIU_PCR139) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 158                  | Pad Configuration Register (SIU_PCR140) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 15A                  | Pad Configuration Register (SIU_PCR141) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 15C                  | Pad Configuration Register (SIU_PCR142) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 15E                  | Pad Configuration Register (SIU_PCR143) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 160                  | Pad Configuration Register (SIU_PCR144) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 162                  | Pad Configuration Register (SIU_PCR145) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 164                  | Pad Configuration Register (SIU_PCR146) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 166                  | Pad Configuration Register (SIU_PCR147) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 168                  | Pad Configuration Register (SIU_PCR148) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 16A                  | Pad Configuration Register (SIU_PCR149) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 16C                  | Pad Configuration Register (SIU_PCR150) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 16E                  | Pad Configuration Register (SIU_PCR151) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 170                  | Pad Configuration Register (SIU_PCR152) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 172                  | Pad Configuration Register (SIU_PCR153) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 174                  | Pad Configuration Register (SIU_PCR154) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 176                  | Pad Configuration Register (SIU_PCR155) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 178                  | Pad Configuration Register (SIU_PCR156) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 17A                  | Pad Configuration Register (SIU_PCR157) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 17C                  | Pad Configuration Register (SIU_PCR158) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 17E                  | Pad Configuration Register (SIU_PCR159) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 180                  | Pad Configuration Register (SIU_PCR160) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 182                  | Pad Configuration Register (SIU_PCR161) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 184                  | Pad Configuration Register (SIU_PCR162) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 186                  | Pad Configuration Register (SIU_PCR163) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 188                  | Pad Configuration Register (SIU_PCR164) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 18A                  | Pad Configuration Register (SIU_PCR165) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 18C                  | Pad Configuration Register (SIU_PCR166) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 18E                  | Pad Configuration Register (SIU_PCR167) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 190                  | Pad Configuration Register (SIU_PCR168) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 192                  | Pad Configuration Register (SIU_PCR169) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 194                  | Pad Configuration Register (SIU_PCR170) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 196                  | Pad Configuration Register (SIU_PCR171) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |

Table continues on the next page...

## SIU memory map (continued)

| Address offset (hex) | Register name                           | Width (in bits) | Access | Reset value                 | Section/ page              |
|----------------------|-----------------------------------------|-----------------|--------|-----------------------------|----------------------------|
| 198                  | Pad Configuration Register (SIU_PCR172) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 19A                  | Pad Configuration Register (SIU_PCR173) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 19C                  | Pad Configuration Register (SIU_PCR174) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 19E                  | Pad Configuration Register (SIU_PCR175) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1A0                  | Pad Configuration Register (SIU_PCR176) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1A2                  | Pad Configuration Register (SIU_PCR177) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1A4                  | Pad Configuration Register (SIU_PCR178) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1A6                  | Pad Configuration Register (SIU_PCR179) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1A8                  | Pad Configuration Register (SIU_PCR180) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1AA                  | Pad Configuration Register (SIU_PCR181) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1AC                  | Pad Configuration Register (SIU_PCR182) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1AE                  | Pad Configuration Register (SIU_PCR183) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1B0                  | Pad Configuration Register (SIU_PCR184) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1B2                  | Pad Configuration Register (SIU_PCR185) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1B4                  | Pad Configuration Register (SIU_PCR186) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1B6                  | Pad Configuration Register (SIU_PCR187) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1B8                  | Pad Configuration Register (SIU_PCR188) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1BA                  | Pad Configuration Register (SIU_PCR189) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1BC                  | Pad Configuration Register (SIU_PCR190) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1BE                  | Pad Configuration Register (SIU_PCR191) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1C0                  | Pad Configuration Register (SIU_PCR192) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1C2                  | Pad Configuration Register (SIU_PCR193) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1C4                  | Pad Configuration Register (SIU_PCR194) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1C6                  | Pad Configuration Register (SIU_PCR195) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1C8                  | Pad Configuration Register (SIU_PCR196) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1CA                  | Pad Configuration Register (SIU_PCR197) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1CC                  | Pad Configuration Register (SIU_PCR198) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1CE                  | Pad Configuration Register (SIU_PCR199) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1D0                  | Pad Configuration Register (SIU_PCR200) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1D2                  | Pad Configuration Register (SIU_PCR201) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1D4                  | Pad Configuration Register (SIU_PCR202) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1D6                  | Pad Configuration Register (SIU_PCR203) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1D8                  | Pad Configuration Register (SIU_PCR204) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1DA                  | Pad Configuration Register (SIU_PCR205) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1DC                  | Pad Configuration Register (SIU_PCR206) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1DE                  | Pad Configuration Register (SIU_PCR207) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1E0                  | Pad Configuration Register (SIU_PCR208) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1E2                  | Pad Configuration Register (SIU_PCR209) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1E4                  | Pad Configuration Register (SIU_PCR210) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |

Table continues on the next page...



## SIU memory map (continued)

| Address offset (hex) | Register name                           | Width (in bits) | Access | Reset value                 | Section/ page              |
|----------------------|-----------------------------------------|-----------------|--------|-----------------------------|----------------------------|
| 1E6                  | Pad Configuration Register (SIU_PCR211) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1E8                  | Pad Configuration Register (SIU_PCR212) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1EA                  | Pad Configuration Register (SIU_PCR213) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1EC                  | Pad Configuration Register (SIU_PCR214) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1EE                  | Pad Configuration Register (SIU_PCR215) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1F0                  | Pad Configuration Register (SIU_PCR216) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1F2                  | Pad Configuration Register (SIU_PCR217) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1F4                  | Pad Configuration Register (SIU_PCR218) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1F6                  | Pad Configuration Register (SIU_PCR219) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1F8                  | Pad Configuration Register (SIU_PCR220) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1FA                  | Pad Configuration Register (SIU_PCR221) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1FC                  | Pad Configuration Register (SIU_PCR222) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 1FE                  | Pad Configuration Register (SIU_PCR223) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 200                  | Pad Configuration Register (SIU_PCR224) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 202                  | Pad Configuration Register (SIU_PCR225) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 204                  | Pad Configuration Register (SIU_PCR226) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 206                  | Pad Configuration Register (SIU_PCR227) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 208                  | Pad Configuration Register (SIU_PCR228) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 20A                  | Pad Configuration Register (SIU_PCR229) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 20C                  | Pad Configuration Register (SIU_PCR230) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 20E                  | Pad Configuration Register (SIU_PCR231) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 210                  | Pad Configuration Register (SIU_PCR232) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 212                  | Pad Configuration Register (SIU_PCR233) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 214                  | Pad Configuration Register (SIU_PCR234) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 216                  | Pad Configuration Register (SIU_PCR235) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 218                  | Pad Configuration Register (SIU_PCR236) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 21A                  | Pad Configuration Register (SIU_PCR237) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 21C                  | Pad Configuration Register (SIU_PCR238) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 21E                  | Pad Configuration Register (SIU_PCR239) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 220                  | Pad Configuration Register (SIU_PCR240) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 222                  | Pad Configuration Register (SIU_PCR241) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 224                  | Pad Configuration Register (SIU_PCR242) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 226                  | Pad Configuration Register (SIU_PCR243) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 228                  | Pad Configuration Register (SIU_PCR244) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 22A                  | Pad Configuration Register (SIU_PCR245) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 22C                  | Pad Configuration Register (SIU_PCR246) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 22E                  | Pad Configuration Register (SIU_PCR247) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 230                  | Pad Configuration Register (SIU_PCR248) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 232                  | Pad Configuration Register (SIU_PCR249) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |

Table continues on the next page...

## SIU memory map (continued)

| Address offset (hex) | Register name                           | Width (in bits) | Access | Reset value                 | Section/ page              |
|----------------------|-----------------------------------------|-----------------|--------|-----------------------------|----------------------------|
| 234                  | Pad Configuration Register (SIU_PCR250) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 236                  | Pad Configuration Register (SIU_PCR251) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 238                  | Pad Configuration Register (SIU_PCR252) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 23A                  | Pad Configuration Register (SIU_PCR253) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 23C                  | Pad Configuration Register (SIU_PCR254) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 23E                  | Pad Configuration Register (SIU_PCR255) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 240                  | Pad Configuration Register (SIU_PCR256) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 242                  | Pad Configuration Register (SIU_PCR257) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 244                  | Pad Configuration Register (SIU_PCR258) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 246                  | Pad Configuration Register (SIU_PCR259) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 248                  | Pad Configuration Register (SIU_PCR260) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 24A                  | Pad Configuration Register (SIU_PCR261) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 24C                  | Pad Configuration Register (SIU_PCR262) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 24E                  | Pad Configuration Register (SIU_PCR263) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 250                  | Pad Configuration Register (SIU_PCR264) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 252                  | Pad Configuration Register (SIU_PCR265) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 254                  | Pad Configuration Register (SIU_PCR266) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 256                  | Pad Configuration Register (SIU_PCR267) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 258                  | Pad Configuration Register (SIU_PCR268) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 25A                  | Pad Configuration Register (SIU_PCR269) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 25C                  | Pad Configuration Register (SIU_PCR270) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 25E                  | Pad Configuration Register (SIU_PCR271) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 260                  | Pad Configuration Register (SIU_PCR272) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 262                  | Pad Configuration Register (SIU_PCR273) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 264                  | Pad Configuration Register (SIU_PCR274) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 266                  | Pad Configuration Register (SIU_PCR275) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 268                  | Pad Configuration Register (SIU_PCR276) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 26A                  | Pad Configuration Register (SIU_PCR277) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 26C                  | Pad Configuration Register (SIU_PCR278) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 26E                  | Pad Configuration Register (SIU_PCR279) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 270                  | Pad Configuration Register (SIU_PCR280) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 272                  | Pad Configuration Register (SIU_PCR281) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 274                  | Pad Configuration Register (SIU_PCR282) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 276                  | Pad Configuration Register (SIU_PCR283) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 278                  | Pad Configuration Register (SIU_PCR284) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 27A                  | Pad Configuration Register (SIU_PCR285) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 27C                  | Pad Configuration Register (SIU_PCR286) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 27E                  | Pad Configuration Register (SIU_PCR287) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 280                  | Pad Configuration Register (SIU_PCR288) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |

Table continues on the next page...



## SIU memory map (continued)

| Address offset (hex) | Register name                           | Width (in bits) | Access | Reset value                 | Section/ page              |
|----------------------|-----------------------------------------|-----------------|--------|-----------------------------|----------------------------|
| 282                  | Pad Configuration Register (SIU_PCR289) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 284                  | Pad Configuration Register (SIU_PCR290) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 286                  | Pad Configuration Register (SIU_PCR291) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 288                  | Pad Configuration Register (SIU_PCR292) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 28A                  | Pad Configuration Register (SIU_PCR293) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 28C                  | Pad Configuration Register (SIU_PCR294) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 28E                  | Pad Configuration Register (SIU_PCR295) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 290                  | Pad Configuration Register (SIU_PCR296) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 292                  | Pad Configuration Register (SIU_PCR297) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 294                  | Pad Configuration Register (SIU_PCR298) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 296                  | Pad Configuration Register (SIU_PCR299) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 298                  | Pad Configuration Register (SIU_PCR300) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 29A                  | Pad Configuration Register (SIU_PCR301) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 29C                  | Pad Configuration Register (SIU_PCR302) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 29E                  | Pad Configuration Register (SIU_PCR303) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2A0                  | Pad Configuration Register (SIU_PCR304) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2A2                  | Pad Configuration Register (SIU_PCR305) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2A4                  | Pad Configuration Register (SIU_PCR306) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2A6                  | Pad Configuration Register (SIU_PCR307) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2A8                  | Pad Configuration Register (SIU_PCR308) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2AA                  | Pad Configuration Register (SIU_PCR309) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2AC                  | Pad Configuration Register (SIU_PCR310) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2AE                  | Pad Configuration Register (SIU_PCR311) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2B0                  | Pad Configuration Register (SIU_PCR312) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2B2                  | Pad Configuration Register (SIU_PCR313) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2B4                  | Pad Configuration Register (SIU_PCR314) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2B6                  | Pad Configuration Register (SIU_PCR315) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2B8                  | Pad Configuration Register (SIU_PCR316) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2BA                  | Pad Configuration Register (SIU_PCR317) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2BC                  | Pad Configuration Register (SIU_PCR318) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2BE                  | Pad Configuration Register (SIU_PCR319) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2C0                  | Pad Configuration Register (SIU_PCR320) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2C2                  | Pad Configuration Register (SIU_PCR321) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2C4                  | Pad Configuration Register (SIU_PCR322) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2C6                  | Pad Configuration Register (SIU_PCR323) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2C8                  | Pad Configuration Register (SIU_PCR324) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2CA                  | Pad Configuration Register (SIU_PCR325) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2CC                  | Pad Configuration Register (SIU_PCR326) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2CE                  | Pad Configuration Register (SIU_PCR327) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |

Table continues on the next page...

## SIU memory map (continued)

| Address offset (hex) | Register name                           | Width (in bits) | Access | Reset value                 | Section/ page              |
|----------------------|-----------------------------------------|-----------------|--------|-----------------------------|----------------------------|
| 2D0                  | Pad Configuration Register (SIU_PCR328) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2D2                  | Pad Configuration Register (SIU_PCR329) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2D4                  | Pad Configuration Register (SIU_PCR330) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2D6                  | Pad Configuration Register (SIU_PCR331) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2D8                  | Pad Configuration Register (SIU_PCR332) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2DA                  | Pad Configuration Register (SIU_PCR333) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2DC                  | Pad Configuration Register (SIU_PCR334) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2DE                  | Pad Configuration Register (SIU_PCR335) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2E0                  | Pad Configuration Register (SIU_PCR336) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2E2                  | Pad Configuration Register (SIU_PCR337) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2E4                  | Pad Configuration Register (SIU_PCR338) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2E6                  | Pad Configuration Register (SIU_PCR339) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2E8                  | Pad Configuration Register (SIU_PCR340) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2EA                  | Pad Configuration Register (SIU_PCR341) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2EC                  | Pad Configuration Register (SIU_PCR342) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2EE                  | Pad Configuration Register (SIU_PCR343) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2F0                  | Pad Configuration Register (SIU_PCR344) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2F2                  | Pad Configuration Register (SIU_PCR345) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2F4                  | Pad Configuration Register (SIU_PCR346) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2F6                  | Pad Configuration Register (SIU_PCR347) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2F8                  | Pad Configuration Register (SIU_PCR348) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2FA                  | Pad Configuration Register (SIU_PCR349) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2FC                  | Pad Configuration Register (SIU_PCR350) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 2FE                  | Pad Configuration Register (SIU_PCR351) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 300                  | Pad Configuration Register (SIU_PCR352) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 302                  | Pad Configuration Register (SIU_PCR353) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 304                  | Pad Configuration Register (SIU_PCR354) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 306                  | Pad Configuration Register (SIU_PCR355) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 308                  | Pad Configuration Register (SIU_PCR356) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 30A                  | Pad Configuration Register (SIU_PCR357) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 30C                  | Pad Configuration Register (SIU_PCR358) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 30E                  | Pad Configuration Register (SIU_PCR359) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 310                  | Pad Configuration Register (SIU_PCR360) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 312                  | Pad Configuration Register (SIU_PCR361) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 314                  | Pad Configuration Register (SIU_PCR362) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 316                  | Pad Configuration Register (SIU_PCR363) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 318                  | Pad Configuration Register (SIU_PCR364) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 31A                  | Pad Configuration Register (SIU_PCR365) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 31C                  | Pad Configuration Register (SIU_PCR366) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |

Table continues on the next page...

## SIU memory map (continued)

| Address offset (hex) | Register name                           | Width (in bits) | Access | Reset value                 | Section/ page              |
|----------------------|-----------------------------------------|-----------------|--------|-----------------------------|----------------------------|
| 31E                  | Pad Configuration Register (SIU_PCR367) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 320                  | Pad Configuration Register (SIU_PCR368) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 322                  | Pad Configuration Register (SIU_PCR369) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 324                  | Pad Configuration Register (SIU_PCR370) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 326                  | Pad Configuration Register (SIU_PCR371) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 328                  | Pad Configuration Register (SIU_PCR372) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 32A                  | Pad Configuration Register (SIU_PCR373) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 32C                  | Pad Configuration Register (SIU_PCR374) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 32E                  | Pad Configuration Register (SIU_PCR375) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 330                  | Pad Configuration Register (SIU_PCR376) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 332                  | Pad Configuration Register (SIU_PCR377) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 334                  | Pad Configuration Register (SIU_PCR378) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 336                  | Pad Configuration Register (SIU_PCR379) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 338                  | Pad Configuration Register (SIU_PCR380) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 33A                  | Pad Configuration Register (SIU_PCR381) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 33C                  | Pad Configuration Register (SIU_PCR382) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 33E                  | Pad Configuration Register (SIU_PCR383) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 340                  | Pad Configuration Register (SIU_PCR384) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 342                  | Pad Configuration Register (SIU_PCR385) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 344                  | Pad Configuration Register (SIU_PCR386) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 346                  | Pad Configuration Register (SIU_PCR387) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 348                  | Pad Configuration Register (SIU_PCR388) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 34A                  | Pad Configuration Register (SIU_PCR389) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 34C                  | Pad Configuration Register (SIU_PCR390) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 34E                  | Pad Configuration Register (SIU_PCR391) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 350                  | Pad Configuration Register (SIU_PCR392) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 352                  | Pad Configuration Register (SIU_PCR393) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 354                  | Pad Configuration Register (SIU_PCR394) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 356                  | Pad Configuration Register (SIU_PCR395) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 358                  | Pad Configuration Register (SIU_PCR396) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 35A                  | Pad Configuration Register (SIU_PCR397) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 35C                  | Pad Configuration Register (SIU_PCR398) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 35E                  | Pad Configuration Register (SIU_PCR399) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 360                  | Pad Configuration Register (SIU_PCR400) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 362                  | Pad Configuration Register (SIU_PCR401) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 364                  | Pad Configuration Register (SIU_PCR402) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 366                  | Pad Configuration Register (SIU_PCR403) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 368                  | Pad Configuration Register (SIU_PCR404) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 36A                  | Pad Configuration Register (SIU_PCR405) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |

Table continues on the next page...

## SIU memory map (continued)

| Address offset (hex) | Register name                           | Width (in bits) | Access | Reset value                 | Section/ page              |
|----------------------|-----------------------------------------|-----------------|--------|-----------------------------|----------------------------|
| 36C                  | Pad Configuration Register (SIU_PCR406) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 36E                  | Pad Configuration Register (SIU_PCR407) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 370                  | Pad Configuration Register (SIU_PCR408) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 372                  | Pad Configuration Register (SIU_PCR409) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 374                  | Pad Configuration Register (SIU_PCR410) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 376                  | Pad Configuration Register (SIU_PCR411) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 378                  | Pad Configuration Register (SIU_PCR412) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 37A                  | Pad Configuration Register (SIU_PCR413) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 37C                  | Pad Configuration Register (SIU_PCR414) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 37E                  | Pad Configuration Register (SIU_PCR415) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 380                  | Pad Configuration Register (SIU_PCR416) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 382                  | Pad Configuration Register (SIU_PCR417) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 384                  | Pad Configuration Register (SIU_PCR418) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 386                  | Pad Configuration Register (SIU_PCR419) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 388                  | Pad Configuration Register (SIU_PCR420) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 38A                  | Pad Configuration Register (SIU_PCR421) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 38C                  | Pad Configuration Register (SIU_PCR422) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 38E                  | Pad Configuration Register (SIU_PCR423) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 390                  | Pad Configuration Register (SIU_PCR424) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 392                  | Pad Configuration Register (SIU_PCR425) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 394                  | Pad Configuration Register (SIU_PCR426) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 396                  | Pad Configuration Register (SIU_PCR427) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 398                  | Pad Configuration Register (SIU_PCR428) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 39A                  | Pad Configuration Register (SIU_PCR429) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 39C                  | Pad Configuration Register (SIU_PCR430) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 39E                  | Pad Configuration Register (SIU_PCR431) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 3A0                  | Pad Configuration Register (SIU_PCR432) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 3A2                  | Pad Configuration Register (SIU_PCR433) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 3A4                  | Pad Configuration Register (SIU_PCR434) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 3A6                  | Pad Configuration Register (SIU_PCR435) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 3A8                  | Pad Configuration Register (SIU_PCR436) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 3AA                  | Pad Configuration Register (SIU_PCR437) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 3AC                  | Pad Configuration Register (SIU_PCR438) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 3AE                  | Pad Configuration Register (SIU_PCR439) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 3B0                  | Pad Configuration Register (SIU_PCR440) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 3B2                  | Pad Configuration Register (SIU_PCR441) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 3B4                  | Pad Configuration Register (SIU_PCR442) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 3B6                  | Pad Configuration Register (SIU_PCR443) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 3B8                  | Pad Configuration Register (SIU_PCR444) | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |

Table continues on the next page...

## SIU memory map (continued)

| Address offset (hex) | Register name                           | Width (in bits) | Access | Reset value                 | Section/ page |
|----------------------|-----------------------------------------|-----------------|--------|-----------------------------|---------------|
| 3BA                  | Pad Configuration Register (SIU_PCR445) | 16              | R/W    | <a href="#">See section</a> | 8.2.13/241    |
| 3BC                  | Pad Configuration Register (SIU_PCR446) | 16              | R/W    | <a href="#">See section</a> | 8.2.13/241    |
| 3BE                  | Pad Configuration Register (SIU_PCR447) | 16              | R/W    | <a href="#">See section</a> | 8.2.13/241    |
| 3C0                  | Pad Configuration Register (SIU_PCR448) | 16              | R/W    | <a href="#">See section</a> | 8.2.13/241    |
| 3C2                  | Pad Configuration Register (SIU_PCR449) | 16              | R/W    | <a href="#">See section</a> | 8.2.13/241    |
| 3C4                  | Pad Configuration Register (SIU_PCR450) | 16              | R/W    | <a href="#">See section</a> | 8.2.13/241    |
| 3C6                  | Pad Configuration Register (SIU_PCR451) | 16              | R/W    | <a href="#">See section</a> | 8.2.13/241    |
| 3C8                  | Pad Configuration Register (SIU_PCR452) | 16              | R/W    | <a href="#">See section</a> | 8.2.13/241    |
| 3CA                  | Pad Configuration Register (SIU_PCR453) | 16              | R/W    | <a href="#">See section</a> | 8.2.13/241    |
| 3CC                  | Pad Configuration Register (SIU_PCR454) | 16              | R/W    | <a href="#">See section</a> | 8.2.13/241    |
| 3CE                  | Pad Configuration Register (SIU_PCR455) | 16              | R/W    | <a href="#">See section</a> | 8.2.13/241    |
| 3D0                  | Pad Configuration Register (SIU_PCR456) | 16              | R/W    | <a href="#">See section</a> | 8.2.13/241    |
| 3D2                  | Pad Configuration Register (SIU_PCR457) | 16              | R/W    | <a href="#">See section</a> | 8.2.13/241    |
| 3D4                  | Pad Configuration Register (SIU_PCR458) | 16              | R/W    | <a href="#">See section</a> | 8.2.13/241    |
| 3D6                  | Pad Configuration Register (SIU_PCR459) | 16              | R/W    | <a href="#">See section</a> | 8.2.13/241    |
| 3D8                  | Pad Configuration Register (SIU_PCR460) | 16              | R/W    | <a href="#">See section</a> | 8.2.13/241    |
| 3DA                  | Pad Configuration Register (SIU_PCR461) | 16              | R/W    | <a href="#">See section</a> | 8.2.13/241    |
| 3DC                  | Pad Configuration Register (SIU_PCR462) | 16              | R/W    | <a href="#">See section</a> | 8.2.13/241    |
| 3DE                  | Pad Configuration Register (SIU_PCR463) | 16              | R/W    | <a href="#">See section</a> | 8.2.13/241    |
| 3E0                  | Pad Configuration Register (SIU_PCR464) | 16              | R/W    | <a href="#">See section</a> | 8.2.13/241    |
| 3E2                  | Pad Configuration Register (SIU_PCR465) | 16              | R/W    | <a href="#">See section</a> | 8.2.13/241    |
| 3E4                  | Pad Configuration Register (SIU_PCR466) | 16              | R/W    | <a href="#">See section</a> | 8.2.13/241    |
| 3E6                  | Pad Configuration Register (SIU_PCR467) | 16              | R/W    | <a href="#">See section</a> | 8.2.13/241    |
| 3E8                  | Pad Configuration Register (SIU_PCR468) | 16              | R/W    | <a href="#">See section</a> | 8.2.13/241    |
| 3EA                  | Pad Configuration Register (SIU_PCR469) | 16              | R/W    | <a href="#">See section</a> | 8.2.13/241    |
| 3EC                  | Pad Configuration Register (SIU_PCR470) | 16              | R/W    | <a href="#">See section</a> | 8.2.13/241    |
| 3EE                  | Pad Configuration Register (SIU_PCR471) | 16              | R/W    | <a href="#">See section</a> | 8.2.13/241    |
| 3F0                  | Pad Configuration Register (SIU_PCR472) | 16              | R/W    | <a href="#">See section</a> | 8.2.13/241    |
| 3F2                  | Pad Configuration Register (SIU_PCR473) | 16              | R/W    | <a href="#">See section</a> | 8.2.13/241    |
| 3F4                  | Pad Configuration Register (SIU_PCR474) | 16              | R/W    | <a href="#">See section</a> | 8.2.13/241    |
| 3F6                  | Pad Configuration Register (SIU_PCR475) | 16              | R/W    | <a href="#">See section</a> | 8.2.13/241    |
| 3F8                  | Pad Configuration Register (SIU_PCR476) | 16              | R/W    | <a href="#">See section</a> | 8.2.13/241    |
| 3FA                  | Pad Configuration Register (SIU_PCR477) | 16              | R/W    | <a href="#">See section</a> | 8.2.13/241    |
| 3FC                  | Pad Configuration Register (SIU_PCR478) | 16              | R/W    | <a href="#">See section</a> | 8.2.13/241    |
| 3FE                  | Pad Configuration Register (SIU_PCR479) | 16              | R/W    | <a href="#">See section</a> | 8.2.13/241    |
| 400                  | Pad Configuration Register (SIU_PCR480) | 16              | R/W    | <a href="#">See section</a> | 8.2.13/241    |
| 402                  | Pad Configuration Register (SIU_PCR481) | 16              | R/W    | <a href="#">See section</a> | 8.2.13/241    |
| 404                  | Pad Configuration Register (SIU_PCR482) | 16              | R/W    | <a href="#">See section</a> | 8.2.13/241    |
| 406                  | Pad Configuration Register (SIU_PCR483) | 16              | R/W    | <a href="#">See section</a> | 8.2.13/241    |

Table continues on the next page...



## SIU memory map (continued)

| Address offset (hex) | Register name                               | Width (in bits) | Access | Reset value                 | Section/ page              |
|----------------------|---------------------------------------------|-----------------|--------|-----------------------------|----------------------------|
| 408                  | Pad Configuration Register (SIU_PCR484)     | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 40A                  | Pad Configuration Register (SIU_PCR485)     | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 40C                  | Pad Configuration Register (SIU_PCR486)     | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 40E                  | Pad Configuration Register (SIU_PCR487)     | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 410                  | Pad Configuration Register (SIU_PCR488)     | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 412                  | Pad Configuration Register (SIU_PCR489)     | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 414                  | Pad Configuration Register (SIU_PCR490)     | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 416                  | Pad Configuration Register (SIU_PCR491)     | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.2.13/241</a> |
| 64B                  | GPIO Pin Data Output Register (SIU_GPDO75)  | 8               | R/W    | 00h                         | <a href="#">8.2.14/244</a> |
| 64C                  | GPIO Pin Data Output Register (SIU_GPDO76)  | 8               | R/W    | 00h                         | <a href="#">8.2.14/244</a> |
| 64D                  | GPIO Pin Data Output Register (SIU_GPDO77)  | 8               | R/W    | 00h                         | <a href="#">8.2.14/244</a> |
| 64E                  | GPIO Pin Data Output Register (SIU_GPDO78)  | 8               | R/W    | 00h                         | <a href="#">8.2.14/244</a> |
| 64F                  | GPIO Pin Data Output Register (SIU_GPDO79)  | 8               | R/W    | 00h                         | <a href="#">8.2.14/244</a> |
| 650                  | GPIO Pin Data Output Register (SIU_GPDO80)  | 8               | R/W    | 00h                         | <a href="#">8.2.14/244</a> |
| 651                  | GPIO Pin Data Output Register (SIU_GPDO81)  | 8               | R/W    | 00h                         | <a href="#">8.2.14/244</a> |
| 652                  | GPIO Pin Data Output Register (SIU_GPDO82)  | 8               | R/W    | 00h                         | <a href="#">8.2.14/244</a> |
| 653                  | GPIO Pin Data Output Register (SIU_GPDO83)  | 8               | R/W    | 00h                         | <a href="#">8.2.14/244</a> |
| 654                  | GPIO Pin Data Output Register (SIU_GPDO84)  | 8               | R/W    | 00h                         | <a href="#">8.2.14/244</a> |
| 655                  | GPIO Pin Data Output Register (SIU_GPDO85)  | 8               | R/W    | 00h                         | <a href="#">8.2.14/244</a> |
| 656                  | GPIO Pin Data Output Register (SIU_GPDO86)  | 8               | R/W    | 00h                         | <a href="#">8.2.14/244</a> |
| 657                  | GPIO Pin Data Output Register (SIU_GPDO87)  | 8               | R/W    | 00h                         | <a href="#">8.2.14/244</a> |
| 658                  | GPIO Pin Data Output Register (SIU_GPDO88)  | 8               | R/W    | 00h                         | <a href="#">8.2.14/244</a> |
| 659                  | GPIO Pin Data Output Register (SIU_GPDO89)  | 8               | R/W    | 00h                         | <a href="#">8.2.14/244</a> |
| 65A                  | GPIO Pin Data Output Register (SIU_GPDO90)  | 8               | R/W    | 00h                         | <a href="#">8.2.14/244</a> |
| 65B                  | GPIO Pin Data Output Register (SIU_GPDO91)  | 8               | R/W    | 00h                         | <a href="#">8.2.14/244</a> |
| 65C                  | GPIO Pin Data Output Register (SIU_GPDO92)  | 8               | R/W    | 00h                         | <a href="#">8.2.14/244</a> |
| 65D                  | GPIO Pin Data Output Register (SIU_GPDO93)  | 8               | R/W    | 00h                         | <a href="#">8.2.14/244</a> |
| 65E                  | GPIO Pin Data Output Register (SIU_GPDO94)  | 8               | R/W    | 00h                         | <a href="#">8.2.14/244</a> |
| 65F                  | GPIO Pin Data Output Register (SIU_GPDO95)  | 8               | R/W    | 00h                         | <a href="#">8.2.14/244</a> |
| 660                  | GPIO Pin Data Output Register (SIU_GPDO96)  | 8               | R/W    | 00h                         | <a href="#">8.2.14/244</a> |
| 661                  | GPIO Pin Data Output Register (SIU_GPDO97)  | 8               | R/W    | 00h                         | <a href="#">8.2.14/244</a> |
| 662                  | GPIO Pin Data Output Register (SIU_GPDO98)  | 8               | R/W    | 00h                         | <a href="#">8.2.14/244</a> |
| 663                  | GPIO Pin Data Output Register (SIU_GPDO99)  | 8               | R/W    | 00h                         | <a href="#">8.2.14/244</a> |
| 664                  | GPIO Pin Data Output Register (SIU_GPDO100) | 8               | R/W    | 00h                         | <a href="#">8.2.14/244</a> |
| 665                  | GPIO Pin Data Output Register (SIU_GPDO101) | 8               | R/W    | 00h                         | <a href="#">8.2.14/244</a> |
| 666                  | GPIO Pin Data Output Register (SIU_GPDO102) | 8               | R/W    | 00h                         | <a href="#">8.2.14/244</a> |
| 667                  | GPIO Pin Data Output Register (SIU_GPDO103) | 8               | R/W    | 00h                         | <a href="#">8.2.14/244</a> |
| 668                  | GPIO Pin Data Output Register (SIU_GPDO104) | 8               | R/W    | 00h                         | <a href="#">8.2.14/244</a> |
| 669                  | GPIO Pin Data Output Register (SIU_GPDO105) | 8               | R/W    | 00h                         | <a href="#">8.2.14/244</a> |

Table continues on the next page...

## SIU memory map (continued)

| Address offset (hex) | Register name                               | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|---------------------------------------------|-----------------|--------|-------------|----------------------------|
| 66A                  | GPIO Pin Data Output Register (SIU_GPDO106) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 66B                  | GPIO Pin Data Output Register (SIU_GPDO107) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 66C                  | GPIO Pin Data Output Register (SIU_GPDO108) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 66D                  | GPIO Pin Data Output Register (SIU_GPDO109) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 66E                  | GPIO Pin Data Output Register (SIU_GPDO110) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 66F                  | GPIO Pin Data Output Register (SIU_GPDO111) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 670                  | GPIO Pin Data Output Register (SIU_GPDO112) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 671                  | GPIO Pin Data Output Register (SIU_GPDO113) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 672                  | GPIO Pin Data Output Register (SIU_GPDO114) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 673                  | GPIO Pin Data Output Register (SIU_GPDO115) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 674                  | GPIO Pin Data Output Register (SIU_GPDO116) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 675                  | GPIO Pin Data Output Register (SIU_GPDO117) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 676                  | GPIO Pin Data Output Register (SIU_GPDO118) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 677                  | GPIO Pin Data Output Register (SIU_GPDO119) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 678                  | GPIO Pin Data Output Register (SIU_GPDO120) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 679                  | GPIO Pin Data Output Register (SIU_GPDO121) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 67A                  | GPIO Pin Data Output Register (SIU_GPDO122) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 67B                  | GPIO Pin Data Output Register (SIU_GPDO123) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 67C                  | GPIO Pin Data Output Register (SIU_GPDO124) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 67D                  | GPIO Pin Data Output Register (SIU_GPDO125) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 67E                  | GPIO Pin Data Output Register (SIU_GPDO126) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 67F                  | GPIO Pin Data Output Register (SIU_GPDO127) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 680                  | GPIO Pin Data Output Register (SIU_GPDO128) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 681                  | GPIO Pin Data Output Register (SIU_GPDO129) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 682                  | GPIO Pin Data Output Register (SIU_GPDO130) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 683                  | GPIO Pin Data Output Register (SIU_GPDO131) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 684                  | GPIO Pin Data Output Register (SIU_GPDO132) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 685                  | GPIO Pin Data Output Register (SIU_GPDO133) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 686                  | GPIO Pin Data Output Register (SIU_GPDO134) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 687                  | GPIO Pin Data Output Register (SIU_GPDO135) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 688                  | GPIO Pin Data Output Register (SIU_GPDO136) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 689                  | GPIO Pin Data Output Register (SIU_GPDO137) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 68A                  | GPIO Pin Data Output Register (SIU_GPDO138) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 68B                  | GPIO Pin Data Output Register (SIU_GPDO139) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 68C                  | GPIO Pin Data Output Register (SIU_GPDO140) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 68D                  | GPIO Pin Data Output Register (SIU_GPDO141) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 68E                  | GPIO Pin Data Output Register (SIU_GPDO142) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 68F                  | GPIO Pin Data Output Register (SIU_GPDO143) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 690                  | GPIO Pin Data Output Register (SIU_GPDO144) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |

Table continues on the next page...

## SIU memory map (continued)

| Address offset (hex) | Register name                               | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|---------------------------------------------|-----------------|--------|-------------|----------------------------|
| 691                  | GPIO Pin Data Output Register (SIU_GPDO145) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 692                  | GPIO Pin Data Output Register (SIU_GPDO146) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 693                  | GPIO Pin Data Output Register (SIU_GPDO147) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 694                  | GPIO Pin Data Output Register (SIU_GPDO148) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 695                  | GPIO Pin Data Output Register (SIU_GPDO149) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 696                  | GPIO Pin Data Output Register (SIU_GPDO150) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 697                  | GPIO Pin Data Output Register (SIU_GPDO151) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 698                  | GPIO Pin Data Output Register (SIU_GPDO152) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 699                  | GPIO Pin Data Output Register (SIU_GPDO153) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 69A                  | GPIO Pin Data Output Register (SIU_GPDO154) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 69B                  | GPIO Pin Data Output Register (SIU_GPDO155) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 69C                  | GPIO Pin Data Output Register (SIU_GPDO156) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 69D                  | GPIO Pin Data Output Register (SIU_GPDO157) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 69E                  | GPIO Pin Data Output Register (SIU_GPDO158) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 69F                  | GPIO Pin Data Output Register (SIU_GPDO159) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6A0                  | GPIO Pin Data Output Register (SIU_GPDO160) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6A1                  | GPIO Pin Data Output Register (SIU_GPDO161) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6A2                  | GPIO Pin Data Output Register (SIU_GPDO162) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6A3                  | GPIO Pin Data Output Register (SIU_GPDO163) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6A4                  | GPIO Pin Data Output Register (SIU_GPDO164) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6A5                  | GPIO Pin Data Output Register (SIU_GPDO165) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6A6                  | GPIO Pin Data Output Register (SIU_GPDO166) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6A7                  | GPIO Pin Data Output Register (SIU_GPDO167) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6A8                  | GPIO Pin Data Output Register (SIU_GPDO168) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6A9                  | GPIO Pin Data Output Register (SIU_GPDO169) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6AA                  | GPIO Pin Data Output Register (SIU_GPDO170) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6AB                  | GPIO Pin Data Output Register (SIU_GPDO171) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6AC                  | GPIO Pin Data Output Register (SIU_GPDO172) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6AD                  | GPIO Pin Data Output Register (SIU_GPDO173) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6AE                  | GPIO Pin Data Output Register (SIU_GPDO174) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6AF                  | GPIO Pin Data Output Register (SIU_GPDO175) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6B0                  | GPIO Pin Data Output Register (SIU_GPDO176) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6B1                  | GPIO Pin Data Output Register (SIU_GPDO177) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6B2                  | GPIO Pin Data Output Register (SIU_GPDO178) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6B3                  | GPIO Pin Data Output Register (SIU_GPDO179) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6B4                  | GPIO Pin Data Output Register (SIU_GPDO180) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6B5                  | GPIO Pin Data Output Register (SIU_GPDO181) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6B6                  | GPIO Pin Data Output Register (SIU_GPDO182) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6B7                  | GPIO Pin Data Output Register (SIU_GPDO183) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |

Table continues on the next page...



## SIU memory map (continued)

| Address offset (hex) | Register name                               | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|---------------------------------------------|-----------------|--------|-------------|----------------------------|
| 6B8                  | GPIO Pin Data Output Register (SIU_GPDO184) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6B9                  | GPIO Pin Data Output Register (SIU_GPDO185) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6BA                  | GPIO Pin Data Output Register (SIU_GPDO186) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6BB                  | GPIO Pin Data Output Register (SIU_GPDO187) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6BC                  | GPIO Pin Data Output Register (SIU_GPDO188) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6BD                  | GPIO Pin Data Output Register (SIU_GPDO189) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6BE                  | GPIO Pin Data Output Register (SIU_GPDO190) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6BF                  | GPIO Pin Data Output Register (SIU_GPDO191) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6C0                  | GPIO Pin Data Output Register (SIU_GPDO192) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6C1                  | GPIO Pin Data Output Register (SIU_GPDO193) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6C2                  | GPIO Pin Data Output Register (SIU_GPDO194) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6C3                  | GPIO Pin Data Output Register (SIU_GPDO195) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6C4                  | GPIO Pin Data Output Register (SIU_GPDO196) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6C5                  | GPIO Pin Data Output Register (SIU_GPDO197) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6C6                  | GPIO Pin Data Output Register (SIU_GPDO198) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6C7                  | GPIO Pin Data Output Register (SIU_GPDO199) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6C8                  | GPIO Pin Data Output Register (SIU_GPDO200) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6C9                  | GPIO Pin Data Output Register (SIU_GPDO201) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6CA                  | GPIO Pin Data Output Register (SIU_GPDO202) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6CB                  | GPIO Pin Data Output Register (SIU_GPDO203) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6CC                  | GPIO Pin Data Output Register (SIU_GPDO204) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6CD                  | GPIO Pin Data Output Register (SIU_GPDO205) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6CE                  | GPIO Pin Data Output Register (SIU_GPDO206) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6CF                  | GPIO Pin Data Output Register (SIU_GPDO207) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6D0                  | GPIO Pin Data Output Register (SIU_GPDO208) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6D1                  | GPIO Pin Data Output Register (SIU_GPDO209) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6D2                  | GPIO Pin Data Output Register (SIU_GPDO210) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6D3                  | GPIO Pin Data Output Register (SIU_GPDO211) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6D4                  | GPIO Pin Data Output Register (SIU_GPDO212) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6D5                  | GPIO Pin Data Output Register (SIU_GPDO213) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6D6                  | GPIO Pin Data Output Register (SIU_GPDO214) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6D7                  | GPIO Pin Data Output Register (SIU_GPDO215) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6D8                  | GPIO Pin Data Output Register (SIU_GPDO216) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6D9                  | GPIO Pin Data Output Register (SIU_GPDO217) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6DA                  | GPIO Pin Data Output Register (SIU_GPDO218) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6DB                  | GPIO Pin Data Output Register (SIU_GPDO219) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6DC                  | GPIO Pin Data Output Register (SIU_GPDO220) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6DD                  | GPIO Pin Data Output Register (SIU_GPDO221) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6DE                  | GPIO Pin Data Output Register (SIU_GPDO222) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |

Table continues on the next page...

## SIU memory map (continued)

| Address offset (hex) | Register name                               | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|---------------------------------------------|-----------------|--------|-------------|----------------------------|
| 6DF                  | GPIO Pin Data Output Register (SIU_GPDO223) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6E0                  | GPIO Pin Data Output Register (SIU_GPDO224) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6E1                  | GPIO Pin Data Output Register (SIU_GPDO225) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6E2                  | GPIO Pin Data Output Register (SIU_GPDO226) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6E3                  | GPIO Pin Data Output Register (SIU_GPDO227) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6E4                  | GPIO Pin Data Output Register (SIU_GPDO228) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6E5                  | GPIO Pin Data Output Register (SIU_GPDO229) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6E6                  | GPIO Pin Data Output Register (SIU_GPDO230) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6E7                  | GPIO Pin Data Output Register (SIU_GPDO231) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6E8                  | GPIO Pin Data Output Register (SIU_GPDO232) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6E9                  | GPIO Pin Data Output Register (SIU_GPDO233) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6EA                  | GPIO Pin Data Output Register (SIU_GPDO234) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6EB                  | GPIO Pin Data Output Register (SIU_GPDO235) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6EC                  | GPIO Pin Data Output Register (SIU_GPDO236) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6ED                  | GPIO Pin Data Output Register (SIU_GPDO237) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6EE                  | GPIO Pin Data Output Register (SIU_GPDO238) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6EF                  | GPIO Pin Data Output Register (SIU_GPDO239) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6F0                  | GPIO Pin Data Output Register (SIU_GPDO240) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6F1                  | GPIO Pin Data Output Register (SIU_GPDO241) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6F2                  | GPIO Pin Data Output Register (SIU_GPDO242) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6F3                  | GPIO Pin Data Output Register (SIU_GPDO243) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6F4                  | GPIO Pin Data Output Register (SIU_GPDO244) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6F5                  | GPIO Pin Data Output Register (SIU_GPDO245) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6F6                  | GPIO Pin Data Output Register (SIU_GPDO246) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6F7                  | GPIO Pin Data Output Register (SIU_GPDO247) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6F8                  | GPIO Pin Data Output Register (SIU_GPDO248) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6F9                  | GPIO Pin Data Output Register (SIU_GPDO249) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6FA                  | GPIO Pin Data Output Register (SIU_GPDO250) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6FB                  | GPIO Pin Data Output Register (SIU_GPDO251) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6FC                  | GPIO Pin Data Output Register (SIU_GPDO252) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6FD                  | GPIO Pin Data Output Register (SIU_GPDO253) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6FE                  | GPIO Pin Data Output Register (SIU_GPDO254) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 6FF                  | GPIO Pin Data Output Register (SIU_GPDO255) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 700                  | GPIO Pin Data Output Register (SIU_GPDO256) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 701                  | GPIO Pin Data Output Register (SIU_GPDO257) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 702                  | GPIO Pin Data Output Register (SIU_GPDO258) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 703                  | GPIO Pin Data Output Register (SIU_GPDO259) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 704                  | GPIO Pin Data Output Register (SIU_GPDO260) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 705                  | GPIO Pin Data Output Register (SIU_GPDO261) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |

Table continues on the next page...

## SIU memory map (continued)

| Address offset (hex) | Register name                               | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|---------------------------------------------|-----------------|--------|-------------|----------------------------|
| 706                  | GPIO Pin Data Output Register (SIU_GPDO262) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 707                  | GPIO Pin Data Output Register (SIU_GPDO263) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 708                  | GPIO Pin Data Output Register (SIU_GPDO264) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 709                  | GPIO Pin Data Output Register (SIU_GPDO265) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 70A                  | GPIO Pin Data Output Register (SIU_GPDO266) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 70B                  | GPIO Pin Data Output Register (SIU_GPDO267) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 70C                  | GPIO Pin Data Output Register (SIU_GPDO268) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 70D                  | GPIO Pin Data Output Register (SIU_GPDO269) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 70E                  | GPIO Pin Data Output Register (SIU_GPDO270) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 70F                  | GPIO Pin Data Output Register (SIU_GPDO271) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 710                  | GPIO Pin Data Output Register (SIU_GPDO272) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 711                  | GPIO Pin Data Output Register (SIU_GPDO273) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 712                  | GPIO Pin Data Output Register (SIU_GPDO274) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 713                  | GPIO Pin Data Output Register (SIU_GPDO275) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 714                  | GPIO Pin Data Output Register (SIU_GPDO276) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 715                  | GPIO Pin Data Output Register (SIU_GPDO277) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 716                  | GPIO Pin Data Output Register (SIU_GPDO278) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 717                  | GPIO Pin Data Output Register (SIU_GPDO279) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 718                  | GPIO Pin Data Output Register (SIU_GPDO280) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 719                  | GPIO Pin Data Output Register (SIU_GPDO281) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 71A                  | GPIO Pin Data Output Register (SIU_GPDO282) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 71B                  | GPIO Pin Data Output Register (SIU_GPDO283) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 71C                  | GPIO Pin Data Output Register (SIU_GPDO284) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 71D                  | GPIO Pin Data Output Register (SIU_GPDO285) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 71E                  | GPIO Pin Data Output Register (SIU_GPDO286) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 71F                  | GPIO Pin Data Output Register (SIU_GPDO287) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 720                  | GPIO Pin Data Output Register (SIU_GPDO288) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 721                  | GPIO Pin Data Output Register (SIU_GPDO289) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 722                  | GPIO Pin Data Output Register (SIU_GPDO290) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 723                  | GPIO Pin Data Output Register (SIU_GPDO291) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 724                  | GPIO Pin Data Output Register (SIU_GPDO292) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 725                  | GPIO Pin Data Output Register (SIU_GPDO293) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 726                  | GPIO Pin Data Output Register (SIU_GPDO294) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 727                  | GPIO Pin Data Output Register (SIU_GPDO295) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 728                  | GPIO Pin Data Output Register (SIU_GPDO296) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 729                  | GPIO Pin Data Output Register (SIU_GPDO297) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 72A                  | GPIO Pin Data Output Register (SIU_GPDO298) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 72B                  | GPIO Pin Data Output Register (SIU_GPDO299) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 72C                  | GPIO Pin Data Output Register (SIU_GPDO300) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |

Table continues on the next page...

## SIU memory map (continued)

| Address offset (hex) | Register name                               | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|---------------------------------------------|-----------------|--------|-------------|----------------------------|
| 72D                  | GPIO Pin Data Output Register (SIU_GPDO301) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 72E                  | GPIO Pin Data Output Register (SIU_GPDO302) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 72F                  | GPIO Pin Data Output Register (SIU_GPDO303) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 730                  | GPIO Pin Data Output Register (SIU_GPDO304) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 731                  | GPIO Pin Data Output Register (SIU_GPDO305) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 732                  | GPIO Pin Data Output Register (SIU_GPDO306) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 733                  | GPIO Pin Data Output Register (SIU_GPDO307) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 734                  | GPIO Pin Data Output Register (SIU_GPDO308) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 735                  | GPIO Pin Data Output Register (SIU_GPDO309) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 736                  | GPIO Pin Data Output Register (SIU_GPDO310) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 737                  | GPIO Pin Data Output Register (SIU_GPDO311) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 738                  | GPIO Pin Data Output Register (SIU_GPDO312) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 739                  | GPIO Pin Data Output Register (SIU_GPDO313) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 73A                  | GPIO Pin Data Output Register (SIU_GPDO314) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 73B                  | GPIO Pin Data Output Register (SIU_GPDO315) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 73C                  | GPIO Pin Data Output Register (SIU_GPDO316) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 73D                  | GPIO Pin Data Output Register (SIU_GPDO317) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 73E                  | GPIO Pin Data Output Register (SIU_GPDO318) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 73F                  | GPIO Pin Data Output Register (SIU_GPDO319) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 740                  | GPIO Pin Data Output Register (SIU_GPDO320) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 741                  | GPIO Pin Data Output Register (SIU_GPDO321) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 742                  | GPIO Pin Data Output Register (SIU_GPDO322) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 743                  | GPIO Pin Data Output Register (SIU_GPDO323) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 744                  | GPIO Pin Data Output Register (SIU_GPDO324) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 745                  | GPIO Pin Data Output Register (SIU_GPDO325) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 746                  | GPIO Pin Data Output Register (SIU_GPDO326) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 747                  | GPIO Pin Data Output Register (SIU_GPDO327) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 748                  | GPIO Pin Data Output Register (SIU_GPDO328) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 749                  | GPIO Pin Data Output Register (SIU_GPDO329) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 74A                  | GPIO Pin Data Output Register (SIU_GPDO330) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 74B                  | GPIO Pin Data Output Register (SIU_GPDO331) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 74C                  | GPIO Pin Data Output Register (SIU_GPDO332) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 74D                  | GPIO Pin Data Output Register (SIU_GPDO333) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 74E                  | GPIO Pin Data Output Register (SIU_GPDO334) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 74F                  | GPIO Pin Data Output Register (SIU_GPDO335) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 750                  | GPIO Pin Data Output Register (SIU_GPDO336) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 751                  | GPIO Pin Data Output Register (SIU_GPDO337) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 752                  | GPIO Pin Data Output Register (SIU_GPDO338) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 753                  | GPIO Pin Data Output Register (SIU_GPDO339) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |

Table continues on the next page...

## SIU memory map (continued)

| Address offset (hex) | Register name                               | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|---------------------------------------------|-----------------|--------|-------------|----------------------------|
| 754                  | GPIO Pin Data Output Register (SIU_GPDO340) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 755                  | GPIO Pin Data Output Register (SIU_GPDO341) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 756                  | GPIO Pin Data Output Register (SIU_GPDO342) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 757                  | GPIO Pin Data Output Register (SIU_GPDO343) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 758                  | GPIO Pin Data Output Register (SIU_GPDO344) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 759                  | GPIO Pin Data Output Register (SIU_GPDO345) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 75A                  | GPIO Pin Data Output Register (SIU_GPDO346) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 75B                  | GPIO Pin Data Output Register (SIU_GPDO347) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 75C                  | GPIO Pin Data Output Register (SIU_GPDO348) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 75D                  | GPIO Pin Data Output Register (SIU_GPDO349) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 75E                  | GPIO Pin Data Output Register (SIU_GPDO350) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 75F                  | GPIO Pin Data Output Register (SIU_GPDO351) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 760                  | GPIO Pin Data Output Register (SIU_GPDO352) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 761                  | GPIO Pin Data Output Register (SIU_GPDO353) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 762                  | GPIO Pin Data Output Register (SIU_GPDO354) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 763                  | GPIO Pin Data Output Register (SIU_GPDO355) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 764                  | GPIO Pin Data Output Register (SIU_GPDO356) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 765                  | GPIO Pin Data Output Register (SIU_GPDO357) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 766                  | GPIO Pin Data Output Register (SIU_GPDO358) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 767                  | GPIO Pin Data Output Register (SIU_GPDO359) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 768                  | GPIO Pin Data Output Register (SIU_GPDO360) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 769                  | GPIO Pin Data Output Register (SIU_GPDO361) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 76A                  | GPIO Pin Data Output Register (SIU_GPDO362) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 76B                  | GPIO Pin Data Output Register (SIU_GPDO363) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 76C                  | GPIO Pin Data Output Register (SIU_GPDO364) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 76D                  | GPIO Pin Data Output Register (SIU_GPDO365) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 76E                  | GPIO Pin Data Output Register (SIU_GPDO366) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 76F                  | GPIO Pin Data Output Register (SIU_GPDO367) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 770                  | GPIO Pin Data Output Register (SIU_GPDO368) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 771                  | GPIO Pin Data Output Register (SIU_GPDO369) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 772                  | GPIO Pin Data Output Register (SIU_GPDO370) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 773                  | GPIO Pin Data Output Register (SIU_GPDO371) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 774                  | GPIO Pin Data Output Register (SIU_GPDO372) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 775                  | GPIO Pin Data Output Register (SIU_GPDO373) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 776                  | GPIO Pin Data Output Register (SIU_GPDO374) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 777                  | GPIO Pin Data Output Register (SIU_GPDO375) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 778                  | GPIO Pin Data Output Register (SIU_GPDO376) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 779                  | GPIO Pin Data Output Register (SIU_GPDO377) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 77A                  | GPIO Pin Data Output Register (SIU_GPDO378) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |

Table continues on the next page...



## SIU memory map (continued)

| Address offset (hex) | Register name                               | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|---------------------------------------------|-----------------|--------|-------------|----------------------------|
| 77B                  | GPIO Pin Data Output Register (SIU_GPDO379) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 77C                  | GPIO Pin Data Output Register (SIU_GPDO380) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 77D                  | GPIO Pin Data Output Register (SIU_GPDO381) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 77E                  | GPIO Pin Data Output Register (SIU_GPDO382) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 77F                  | GPIO Pin Data Output Register (SIU_GPDO383) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 780                  | GPIO Pin Data Output Register (SIU_GPDO384) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 781                  | GPIO Pin Data Output Register (SIU_GPDO385) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 782                  | GPIO Pin Data Output Register (SIU_GPDO386) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 783                  | GPIO Pin Data Output Register (SIU_GPDO387) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 784                  | GPIO Pin Data Output Register (SIU_GPDO388) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 785                  | GPIO Pin Data Output Register (SIU_GPDO389) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 786                  | GPIO Pin Data Output Register (SIU_GPDO390) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 787                  | GPIO Pin Data Output Register (SIU_GPDO391) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 788                  | GPIO Pin Data Output Register (SIU_GPDO392) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 789                  | GPIO Pin Data Output Register (SIU_GPDO393) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 78A                  | GPIO Pin Data Output Register (SIU_GPDO394) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 78B                  | GPIO Pin Data Output Register (SIU_GPDO395) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 78C                  | GPIO Pin Data Output Register (SIU_GPDO396) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 78D                  | GPIO Pin Data Output Register (SIU_GPDO397) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 78E                  | GPIO Pin Data Output Register (SIU_GPDO398) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 78F                  | GPIO Pin Data Output Register (SIU_GPDO399) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 790                  | GPIO Pin Data Output Register (SIU_GPDO400) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 791                  | GPIO Pin Data Output Register (SIU_GPDO401) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 792                  | GPIO Pin Data Output Register (SIU_GPDO402) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 793                  | GPIO Pin Data Output Register (SIU_GPDO403) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 794                  | GPIO Pin Data Output Register (SIU_GPDO404) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 795                  | GPIO Pin Data Output Register (SIU_GPDO405) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 796                  | GPIO Pin Data Output Register (SIU_GPDO406) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 797                  | GPIO Pin Data Output Register (SIU_GPDO407) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 798                  | GPIO Pin Data Output Register (SIU_GPDO408) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 799                  | GPIO Pin Data Output Register (SIU_GPDO409) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 79A                  | GPIO Pin Data Output Register (SIU_GPDO410) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 79B                  | GPIO Pin Data Output Register (SIU_GPDO411) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 79C                  | GPIO Pin Data Output Register (SIU_GPDO412) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 79D                  | GPIO Pin Data Output Register (SIU_GPDO413) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 79E                  | GPIO Pin Data Output Register (SIU_GPDO414) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 79F                  | GPIO Pin Data Output Register (SIU_GPDO415) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7A0                  | GPIO Pin Data Output Register (SIU_GPDO416) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7A1                  | GPIO Pin Data Output Register (SIU_GPDO417) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |

Table continues on the next page...

## SIU memory map (continued)

| Address offset (hex) | Register name                               | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|---------------------------------------------|-----------------|--------|-------------|----------------------------|
| 7A2                  | GPIO Pin Data Output Register (SIU_GPDO418) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7A3                  | GPIO Pin Data Output Register (SIU_GPDO419) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7A4                  | GPIO Pin Data Output Register (SIU_GPDO420) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7A5                  | GPIO Pin Data Output Register (SIU_GPDO421) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7A6                  | GPIO Pin Data Output Register (SIU_GPDO422) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7A7                  | GPIO Pin Data Output Register (SIU_GPDO423) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7A8                  | GPIO Pin Data Output Register (SIU_GPDO424) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7A9                  | GPIO Pin Data Output Register (SIU_GPDO425) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7AA                  | GPIO Pin Data Output Register (SIU_GPDO426) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7AB                  | GPIO Pin Data Output Register (SIU_GPDO427) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7AC                  | GPIO Pin Data Output Register (SIU_GPDO428) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7AD                  | GPIO Pin Data Output Register (SIU_GPDO429) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7AE                  | GPIO Pin Data Output Register (SIU_GPDO430) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7AF                  | GPIO Pin Data Output Register (SIU_GPDO431) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7B0                  | GPIO Pin Data Output Register (SIU_GPDO432) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7B1                  | GPIO Pin Data Output Register (SIU_GPDO433) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7B2                  | GPIO Pin Data Output Register (SIU_GPDO434) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7B3                  | GPIO Pin Data Output Register (SIU_GPDO435) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7B4                  | GPIO Pin Data Output Register (SIU_GPDO436) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7B5                  | GPIO Pin Data Output Register (SIU_GPDO437) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7B6                  | GPIO Pin Data Output Register (SIU_GPDO438) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7B7                  | GPIO Pin Data Output Register (SIU_GPDO439) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7B8                  | GPIO Pin Data Output Register (SIU_GPDO440) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7B9                  | GPIO Pin Data Output Register (SIU_GPDO441) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7BA                  | GPIO Pin Data Output Register (SIU_GPDO442) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7BB                  | GPIO Pin Data Output Register (SIU_GPDO443) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7BC                  | GPIO Pin Data Output Register (SIU_GPDO444) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7BD                  | GPIO Pin Data Output Register (SIU_GPDO445) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7BE                  | GPIO Pin Data Output Register (SIU_GPDO446) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7BF                  | GPIO Pin Data Output Register (SIU_GPDO447) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7C0                  | GPIO Pin Data Output Register (SIU_GPDO448) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7C1                  | GPIO Pin Data Output Register (SIU_GPDO449) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7C2                  | GPIO Pin Data Output Register (SIU_GPDO450) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7C3                  | GPIO Pin Data Output Register (SIU_GPDO451) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7C4                  | GPIO Pin Data Output Register (SIU_GPDO452) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7C5                  | GPIO Pin Data Output Register (SIU_GPDO453) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7C6                  | GPIO Pin Data Output Register (SIU_GPDO454) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7C7                  | GPIO Pin Data Output Register (SIU_GPDO455) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7C8                  | GPIO Pin Data Output Register (SIU_GPDO456) | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |

Table continues on the next page...

## SIU memory map (continued)

| Address offset (hex) | Register name                                       | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|-----------------------------------------------------|-----------------|--------|-------------|----------------------------|
| 7C9                  | GPIO Pin Data Output Register (SIU_GPDO457)         | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7CA                  | GPIO Pin Data Output Register (SIU_GPDO458)         | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7CB                  | GPIO Pin Data Output Register (SIU_GPDO459)         | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7CC                  | GPIO Pin Data Output Register (SIU_GPDO460)         | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7CD                  | GPIO Pin Data Output Register (SIU_GPDO461)         | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7CE                  | GPIO Pin Data Output Register (SIU_GPDO462)         | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7CF                  | GPIO Pin Data Output Register (SIU_GPDO463)         | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7D0                  | GPIO Pin Data Output Register (SIU_GPDO464)         | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7D1                  | GPIO Pin Data Output Register (SIU_GPDO465)         | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7D2                  | GPIO Pin Data Output Register (SIU_GPDO466)         | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7D3                  | GPIO Pin Data Output Register (SIU_GPDO467)         | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7D4                  | GPIO Pin Data Output Register (SIU_GPDO468)         | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7D5                  | GPIO Pin Data Output Register (SIU_GPDO469)         | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7D6                  | GPIO Pin Data Output Register (SIU_GPDO470)         | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7D7                  | GPIO Pin Data Output Register (SIU_GPDO471)         | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7D8                  | GPIO Pin Data Output Register (SIU_GPDO472)         | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7D9                  | GPIO Pin Data Output Register (SIU_GPDO473)         | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7DA                  | GPIO Pin Data Output Register (SIU_GPDO474)         | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7DB                  | GPIO Pin Data Output Register (SIU_GPDO475)         | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7DC                  | GPIO Pin Data Output Register (SIU_GPDO476)         | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7DD                  | GPIO Pin Data Output Register (SIU_GPDO477)         | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7DE                  | GPIO Pin Data Output Register (SIU_GPDO478)         | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7DF                  | GPIO Pin Data Output Register (SIU_GPDO479)         | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7E0                  | GPIO Pin Data Output Register (SIU_GPDO480)         | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7E1                  | GPIO Pin Data Output Register (SIU_GPDO481)         | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7E2                  | GPIO Pin Data Output Register (SIU_GPDO482)         | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7E3                  | GPIO Pin Data Output Register (SIU_GPDO483)         | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7E4                  | GPIO Pin Data Output Register (SIU_GPDO484)         | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7E5                  | GPIO Pin Data Output Register (SIU_GPDO485)         | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7E6                  | GPIO Pin Data Output Register (SIU_GPDO486)         | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7E7                  | GPIO Pin Data Output Register (SIU_GPDO487)         | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7E8                  | GPIO Pin Data Output Register (SIU_GPDO488)         | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7E9                  | GPIO Pin Data Output Register (SIU_GPDO489)         | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7EA                  | GPIO Pin Data Output Register (SIU_GPDO490)         | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 7EB                  | GPIO Pin Data Output Register (SIU_GPDO491)         | 8               | R/W    | 00h         | <a href="#">8.2.14/244</a> |
| 84B                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI75L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 84C                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI76L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 84D                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI77L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 84E                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI78L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |

Table continues on the next page...



## SIU memory map (continued)

| Address offset (hex) | Register name                                        | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|------------------------------------------------------|-----------------|--------|-------------|----------------------------|
| 84F                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI79L)  | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 850                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI80L)  | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 851                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI81L)  | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 852                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI82L)  | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 853                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI83L)  | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 854                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI84L)  | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 855                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI85L)  | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 856                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI86L)  | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 857                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI87L)  | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 858                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI88L)  | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 859                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI89L)  | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 85A                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI90L)  | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 85B                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI91L)  | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 85C                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI92L)  | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 85D                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI93L)  | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 85E                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI94L)  | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 85F                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI95L)  | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 860                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI96L)  | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 861                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI97L)  | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 862                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI98L)  | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 863                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI99L)  | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 864                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI100L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 865                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI101L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 866                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI102L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 867                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI103L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 868                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI104L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 869                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI105L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 86A                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI106L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 86B                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI107L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 86C                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI108L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 86D                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI109L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 86E                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI110L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 86F                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI111L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 870                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI112L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 871                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI113L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 872                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI114L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 873                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI115L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 874                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI116L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 875                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI117L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |

Table continues on the next page...

## SIU memory map (continued)

| Address offset (hex) | Register name                                        | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|------------------------------------------------------|-----------------|--------|-------------|----------------------------|
| 876                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI118L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 877                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI119L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 878                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI120L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 879                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI121L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 87A                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI122L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 87B                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI123L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 87C                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI124L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 87D                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI125L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 87E                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI126L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 87F                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI127L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 880                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI128L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 881                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI129L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 882                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI130L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 883                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI131L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 884                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI132L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 885                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI133L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 886                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI134L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 887                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI135L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 888                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI136L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 889                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI137L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 88A                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI138L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 88B                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI139L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 88C                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI140L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 88D                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI141L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 88E                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI142L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 88F                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI143L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 890                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI144L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 891                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI145L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 892                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI146L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 893                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI147L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 894                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI148L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 895                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI149L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 896                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI150L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 897                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI151L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 898                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI152L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 899                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI153L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 89A                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI154L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 89B                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI155L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 89C                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI156L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |

Table continues on the next page...

## SIU memory map (continued)

| Address offset (hex) | Register name                                        | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|------------------------------------------------------|-----------------|--------|-------------|----------------------------|
| 89D                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI157L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 89E                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI158L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 89F                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI159L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8A0                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI160L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8A1                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI161L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8A2                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI162L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8A3                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI163L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8A4                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI164L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8A5                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI165L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8A6                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI166L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8A7                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI167L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8A8                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI168L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8A9                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI169L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8AA                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI170L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8AB                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI171L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8AC                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI172L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8AD                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI173L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8AE                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI174L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8AF                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI175L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8B0                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI176L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8B1                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI177L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8B2                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI178L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8B3                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI179L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8B4                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI180L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8B5                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI181L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8B6                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI182L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8B7                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI183L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8B8                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI184L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8B9                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI185L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8BA                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI186L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8BB                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI187L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8BC                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI188L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8BD                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI189L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8BE                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI190L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8BF                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI191L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8C0                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI192L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8C1                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI193L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8C2                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI194L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8C3                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI195L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |

Table continues on the next page...

## SIU memory map (continued)

| Address offset (hex) | Register name                                        | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|------------------------------------------------------|-----------------|--------|-------------|----------------------------|
| 8C4                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI196L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8C5                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI197L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8C6                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI198L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8C7                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI199L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8C8                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI200L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8C9                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI201L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8CA                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI202L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8CB                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI203L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8CC                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI204L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8CD                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI205L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8CE                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI206L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8CF                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI207L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8D0                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI208L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8D1                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI209L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8D2                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI210L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8D3                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI211L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8D4                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI212L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8D5                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI213L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8D6                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI214L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8D7                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI215L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8D8                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI216L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8D9                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI217L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8DA                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI218L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8DB                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI219L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8DC                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI220L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8DD                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI221L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8DE                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI222L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8DF                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI223L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8E0                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI224L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8E1                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI225L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8E2                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI226L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8E3                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI227L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8E4                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI228L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8E5                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI229L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8E6                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI230L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8E7                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI231L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8E8                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI232L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8E9                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI233L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8EA                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI234L) | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |

Table continues on the next page...

## SIU memory map (continued)

| Address offset (hex) | Register name                                                         | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|-----------------------------------------------------------------------|-----------------|--------|-------------|----------------------------|
| 8EB                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI235L)                  | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8EC                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI236L)                  | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8ED                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI237L)                  | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8EE                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI238L)                  | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8EF                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI239L)                  | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8F0                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI240L)                  | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8F1                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI241L)                  | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8F2                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI242L)                  | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8F3                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI243L)                  | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8F4                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI244L)                  | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8F5                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI245L)                  | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8F6                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI246L)                  | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8F7                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI247L)                  | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8F8                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI248L)                  | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8F9                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI249L)                  | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8FA                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI250L)                  | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8FB                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI251L)                  | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8FC                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI252L)                  | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 8FD                  | GPIO Pin Data Input Register (legacy) (SIU_GPDI253L)                  | 8               | R      | 00h         | <a href="#">8.2.15/245</a> |
| 904                  | External IRQ Input Select Register (SIU_EIISR)                        | 32              | R/W    | 0000_0000h  | <a href="#">8.2.16/246</a> |
| 908                  | DSPI Input Select Register (SIU_DISR)                                 | 32              | R/W    | 0000_0000h  | <a href="#">8.2.17/248</a> |
| 910                  | eQADC Command FIFO Trigger Source Registers (SIU_ISEL4)               | 32              | R/W    | 0000_0000h  | <a href="#">8.2.18/250</a> |
| 914                  | eQADC Command FIFO Trigger Source Registers (SIU_ISEL5)               | 32              | R/W    | 0000_0000h  | <a href="#">8.2.19/255</a> |
| 918                  | eQADC Command FIFO Trigger Source Registers (SIU_ISEL6)               | 32              | R/W    | 0000_0000h  | <a href="#">8.2.20/258</a> |
| 91C                  | eQADC Command FIFO Trigger Source Registers (SIU_ISEL7)               | 32              | R/W    | 0000_0000h  | <a href="#">8.2.21/262</a> |
| 920                  | eTPU_A Input Select Register (SIU_ISEL8)                              | 32              | R/W    | 0000_0000h  | <a href="#">8.2.22/265</a> |
| 924                  | eQADC Advance Trigger Source Register (SIU_ISEL9)                     | 32              | R/W    | 0000_0000h  | <a href="#">8.2.23/266</a> |
| 940                  | Sigma-Delta Gate Selection Register (SIU_SDGATE_SEL)                  | 32              | R/W    | 0000_0000h  | <a href="#">8.2.24/269</a> |
| 944                  | Sigma-Delta to eTPU_A Connection Configuration Register (SIU_SDETPUA) | 32              | R/W    | 0000_0000h  | <a href="#">8.2.25/270</a> |
| 948                  | Sigma-Delta to eTPU_B Connection Configuration Register (SIU_SDETPUB) | 32              | R/W    | 0000_0000h  | <a href="#">8.2.26/273</a> |
| 94C                  | Sigma-Delta to eTPU_C Connection Configuration Register (SIU_SDETPUC) | 32              | R/W    | 0000_0000h  | <a href="#">8.2.27/275</a> |
| 950                  | Trigger Burst Generator Control Registers for eQADC_A (SIU_TBG_CR_A0) | 32              | R/W    | 0000_0000h  | <a href="#">8.2.28/278</a> |

Table continues on the next page...



## SIU memory map (continued)

| Address offset (hex) | Register name                                                         | Width (in bits) | Access | Reset value                 | Section/ page              |
|----------------------|-----------------------------------------------------------------------|-----------------|--------|-----------------------------|----------------------------|
| 954                  | Trigger Burst Generator Control Registers for eQADC_A (SIU_TBG_CR_A1) | 32              | R/W    | 0000_0000h                  | <a href="#">8.2.28/278</a> |
| 958                  | Trigger Burst Generator Control Registers for eQADC_A (SIU_TBG_CR_A2) | 32              | R/W    | 0000_0000h                  | <a href="#">8.2.28/278</a> |
| 95C                  | Trigger Burst Generator Control Registers for eQADC_A (SIU_TBG_CR_A3) | 32              | R/W    | 0000_0000h                  | <a href="#">8.2.28/278</a> |
| 960                  | Trigger Burst Generator Control Registers for eQADC_A (SIU_TBG_CR_A4) | 32              | R/W    | 0000_0000h                  | <a href="#">8.2.28/278</a> |
| 964                  | Trigger Burst Generator Control Registers for eQADC_A (SIU_TBG_CR_A5) | 32              | R/W    | 0000_0000h                  | <a href="#">8.2.28/278</a> |
| 968                  | Trigger Burst Generator Control Registers for eQADC_B (SIU_TBG_CR_B0) | 32              | R/W    | 0000_0000h                  | <a href="#">8.2.29/279</a> |
| 96C                  | Trigger Burst Generator Control Registers for eQADC_B (SIU_TBG_CR_B1) | 32              | R/W    | 0000_0000h                  | <a href="#">8.2.29/279</a> |
| 970                  | Trigger Burst Generator Control Registers for eQADC_B (SIU_TBG_CR_B2) | 32              | R/W    | 0000_0000h                  | <a href="#">8.2.29/279</a> |
| 974                  | Trigger Burst Generator Control Registers for eQADC_B (SIU_TBG_CR_B3) | 32              | R/W    | 0000_0000h                  | <a href="#">8.2.29/279</a> |
| 978                  | Trigger Burst Generator Control Registers for eQADC_B (SIU_TBG_CR_B4) | 32              | R/W    | 0000_0000h                  | <a href="#">8.2.29/279</a> |
| 97C                  | Trigger Burst Generator Control Registers for eQADC_B (SIU_TBG_CR_B5) | 32              | R/W    | 0000_0000h                  | <a href="#">8.2.29/279</a> |
| 984                  | External Clock Control Register (SIU_ECCR)                            | 32              | R/W    | 0000_1001h                  | <a href="#">8.2.30/280</a> |
| 9A0                  | System Clock Register (SIU_SYSDIV)                                    | 32              | R/W    | 0000_0010h                  | <a href="#">8.2.31/281</a> |
| 9A4                  | Halt Register 1 (SIU_HLT1)                                            | 32              | R/W    | 4000_0000h                  | <a href="#">8.2.32/284</a> |
| 9A8                  | Halt Acknowledge Register 1 (SIU_HLTACK1)                             | 32              | R      | 0000_0000h                  | <a href="#">8.2.33/288</a> |
| 9AC                  | Core0 Reset Vector Register (SIU_RSTVEC0)                             | 32              | R/W    | FFFF_FFFCh                  | <a href="#">8.2.34/291</a> |
| 9B0                  | Core1 Reset Vector Register (SIU_RSTVEC1)                             | 32              | R/W    | FFFF_FFFEh                  | <a href="#">8.2.35/292</a> |
| 9B4                  | Core0 PID mapping control register (SIU_C0PID)                        | 32              | R/W    | 0000_0000h                  | <a href="#">8.2.36/294</a> |
| 9B8                  | Core1 PID mapping control register (SIU_C1PID)                        | 32              | R/W    | 0000_0000h                  | <a href="#">8.2.37/295</a> |
| 9BC                  | Halt Register 2 (SIU_HLT2)                                            | 32              | R/W    | 0000_0000h                  | <a href="#">8.2.38/296</a> |
| 9C0                  | Halt Acknowledge Register 2 (SIU_HLTACK2)                             | 32              | R      | 0000_0000h                  | <a href="#">8.2.39/299</a> |
| 9C4                  | Sigma-Delta Clock Configuration Register (SIU_SDCLKCFG)               | 32              | R/W    | 0000_0000h                  | <a href="#">8.2.40/301</a> |
| 9D0                  | Reset Configuration Register (SIU_RCR)                                | 32              | R/W    | <a href="#">See section</a> | <a href="#">8.2.41/302</a> |
| 9D4                  | Lock-Step Status Register (SIU_LOCKSTEP)                              | 32              | R      | 0000_0000h                  | <a href="#">8.2.42/302</a> |
| 9D8                  | Progressive Clock Switch Interrupt Enable Register (SIU_PCSEIR)       | 32              | R/W    | 0000_0000h                  | <a href="#">8.2.43/303</a> |
| 9DC                  | Progressive Clock Switch Interrupt Flag Register (SIU_PCSIFR)         | 32              | R/W    | 0000_0000h                  | <a href="#">8.2.44/304</a> |
| 9E0                  | Fast Ethernet Controller Control Register (SIU_FECCR)                 | 32              | R/W    | 0000_0000h                  | <a href="#">8.2.45/305</a> |
| 9E4                  | ECC Error Injection Register (SIU_ECCEIR)                             | 32              | R/W    | 0000_0000h                  | <a href="#">8.2.46/305</a> |
| C00                  | Parallel GPIO Pin Data Output Registers (SIU_PGPDO0)                  | 32              | R/W    | 0000_0000h                  | <a href="#">8.2.47/306</a> |

Table continues on the next page...

## SIU memory map (continued)

| Address offset (hex) | Register name                                                | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|--------------------------------------------------------------|-----------------|--------|-------------|----------------------------|
| C04                  | Parallel GPIO Pin Data Output Registers (SIU_PGPDO1)         | 32              | R/W    | 0000_0000h  | <a href="#">8.2.47/306</a> |
| C08                  | Parallel GPIO Pin Data Output Registers (SIU_PGPDO2)         | 32              | R/W    | 0000_0000h  | <a href="#">8.2.47/306</a> |
| C0C                  | Parallel GPIO Pin Data Output Registers (SIU_PGPDO3)         | 32              | R/W    | 0000_0000h  | <a href="#">8.2.47/306</a> |
| C10                  | Parallel GPIO Pin Data Output Registers (SIU_PGPDO4)         | 32              | R/W    | 0000_0000h  | <a href="#">8.2.47/306</a> |
| C14                  | Parallel GPIO Pin Data Output Registers (SIU_PGPDO5)         | 32              | R/W    | 0000_0000h  | <a href="#">8.2.47/306</a> |
| C18                  | Parallel GPIO Pin Data Output Registers (SIU_PGPDO6)         | 32              | R/W    | 0000_0000h  | <a href="#">8.2.47/306</a> |
| C1C                  | Parallel GPIO Pin Data Output Registers (SIU_PGPDO7)         | 32              | R/W    | 0000_0000h  | <a href="#">8.2.47/306</a> |
| C20                  | Parallel GPIO Pin Data Output Registers (SIU_PGPDO8)         | 32              | R/W    | 0000_0000h  | <a href="#">8.2.47/306</a> |
| C24                  | Parallel GPIO Pin Data Output Registers (SIU_PGPDO9)         | 32              | R/W    | 0000_0000h  | <a href="#">8.2.47/306</a> |
| C28                  | Parallel GPIO Pin Data Output Registers (SIU_PGPDO10)        | 32              | R/W    | 0000_0000h  | <a href="#">8.2.47/306</a> |
| C2C                  | Parallel GPIO Pin Data Output Registers (SIU_PGPDO11)        | 32              | R/W    | 0000_0000h  | <a href="#">8.2.47/306</a> |
| C30                  | Parallel GPIO Pin Data Output Registers (SIU_PGPDO12)        | 32              | R/W    | 0000_0000h  | <a href="#">8.2.47/306</a> |
| C34                  | Parallel GPIO Pin Data Output Registers (SIU_PGPDO13)        | 32              | R/W    | 0000_0000h  | <a href="#">8.2.47/306</a> |
| C38                  | Parallel GPIO Pin Data Output Registers (SIU_PGPDO14)        | 32              | R/W    | 0000_0000h  | <a href="#">8.2.47/306</a> |
| C3C                  | Parallel GPIO Pin Data Output Registers (SIU_PGPDO15)        | 32              | R/W    | 0000_0000h  | <a href="#">8.2.47/306</a> |
| C40                  | Parallel GPIO Pin Data Input Registers (SIU_PGPD10)          | 32              | R      | 0000_0000h  | <a href="#">8.2.48/310</a> |
| C44                  | Parallel GPIO Pin Data Input Registers (SIU_PGPD11)          | 32              | R      | 0000_0000h  | <a href="#">8.2.48/310</a> |
| C48                  | Parallel GPIO Pin Data Input Registers (SIU_PGPD12)          | 32              | R      | 0000_0000h  | <a href="#">8.2.48/310</a> |
| C4C                  | Parallel GPIO Pin Data Input Registers (SIU_PGPD13)          | 32              | R      | 0000_0000h  | <a href="#">8.2.48/310</a> |
| C50                  | Parallel GPIO Pin Data Input Registers (SIU_PGPD14)          | 32              | R      | 0000_0000h  | <a href="#">8.2.48/310</a> |
| C54                  | Parallel GPIO Pin Data Input Registers (SIU_PGPD15)          | 32              | R      | 0000_0000h  | <a href="#">8.2.48/310</a> |
| C58                  | Parallel GPIO Pin Data Input Registers (SIU_PGPD16)          | 32              | R      | 0000_0000h  | <a href="#">8.2.48/310</a> |
| C5C                  | Parallel GPIO Pin Data Input Registers (SIU_PGPD17)          | 32              | R      | 0000_0000h  | <a href="#">8.2.48/310</a> |
| C60                  | Parallel GPIO Pin Data Input Registers (SIU_PGPD18)          | 32              | R      | 0000_0000h  | <a href="#">8.2.48/310</a> |
| C64                  | Parallel GPIO Pin Data Input Registers (SIU_PGPD19)          | 32              | R      | 0000_0000h  | <a href="#">8.2.48/310</a> |
| C68                  | Parallel GPIO Pin Data Input Registers (SIU_PGPD10)          | 32              | R      | 0000_0000h  | <a href="#">8.2.48/310</a> |
| C6C                  | Parallel GPIO Pin Data Input Registers (SIU_PGPD11)          | 32              | R      | 0000_0000h  | <a href="#">8.2.48/310</a> |
| C70                  | Parallel GPIO Pin Data Input Registers (SIU_PGPD12)          | 32              | R      | 0000_0000h  | <a href="#">8.2.48/310</a> |
| C74                  | Parallel GPIO Pin Data Input Registers (SIU_PGPD13)          | 32              | R      | 0000_0000h  | <a href="#">8.2.48/310</a> |
| C78                  | Parallel GPIO Pin Data Input Registers (SIU_PGPD14)          | 32              | R      | 0000_0000h  | <a href="#">8.2.48/310</a> |
| C7C                  | Parallel GPIO Pin Data Input Registers (SIU_PGPD15)          | 32              | R      | 0000_0000h  | <a href="#">8.2.48/310</a> |
| C80                  | Masked Parallel GPIO Pin Data Output Registers (SIU_MPGPDO0) | 32              | W      | 0000_0000h  | <a href="#">8.2.49/314</a> |
| C84                  | Masked Parallel GPIO Pin Data Output Registers (SIU_MPGPDO1) | 32              | W      | 0000_0000h  | <a href="#">8.2.49/314</a> |
| C88                  | Masked Parallel GPIO Pin Data Output Registers (SIU_MPGPDO2) | 32              | W      | 0000_0000h  | <a href="#">8.2.49/314</a> |
| C8C                  | Masked Parallel GPIO Pin Data Output Registers (SIU_MPGPDO3) | 32              | W      | 0000_0000h  | <a href="#">8.2.49/314</a> |

Table continues on the next page...

## SIU memory map (continued)

| Address offset (hex) | Register name                                                 | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|---------------------------------------------------------------|-----------------|--------|-------------|----------------------------|
| C90                  | Masked Parallel GPIO Pin Data Output Registers (SIU_MPGPDO4)  | 32              | W      | 0000_0000h  | <a href="#">8.2.49/314</a> |
| C94                  | Masked Parallel GPIO Pin Data Output Registers (SIU_MPGPDO5)  | 32              | W      | 0000_0000h  | <a href="#">8.2.49/314</a> |
| C98                  | Masked Parallel GPIO Pin Data Output Registers (SIU_MPGPDO6)  | 32              | W      | 0000_0000h  | <a href="#">8.2.49/314</a> |
| C9C                  | Masked Parallel GPIO Pin Data Output Registers (SIU_MPGPDO7)  | 32              | W      | 0000_0000h  | <a href="#">8.2.49/314</a> |
| CA0                  | Masked Parallel GPIO Pin Data Output Registers (SIU_MPGPDO8)  | 32              | W      | 0000_0000h  | <a href="#">8.2.49/314</a> |
| CA4                  | Masked Parallel GPIO Pin Data Output Registers (SIU_MPGPDO9)  | 32              | W      | 0000_0000h  | <a href="#">8.2.49/314</a> |
| CA8                  | Masked Parallel GPIO Pin Data Output Registers (SIU_MPGPDO10) | 32              | W      | 0000_0000h  | <a href="#">8.2.49/314</a> |
| CAC                  | Masked Parallel GPIO Pin Data Output Registers (SIU_MPGPDO11) | 32              | W      | 0000_0000h  | <a href="#">8.2.49/314</a> |
| CB0                  | Masked Parallel GPIO Pin Data Output Registers (SIU_MPGPDO12) | 32              | W      | 0000_0000h  | <a href="#">8.2.49/314</a> |
| CB4                  | Masked Parallel GPIO Pin Data Output Registers (SIU_MPGPDO13) | 32              | W      | 0000_0000h  | <a href="#">8.2.49/314</a> |
| CB8                  | Masked Parallel GPIO Pin Data Output Registers (SIU_MPGPDO14) | 32              | W      | 0000_0000h  | <a href="#">8.2.49/314</a> |
| CBC                  | Masked Parallel GPIO Pin Data Output Registers (SIU_MPGPDO15) | 32              | W      | 0000_0000h  | <a href="#">8.2.49/314</a> |
| CC0                  | Masked Parallel GPIO Pin Data Output Registers (SIU_MPGPDO16) | 32              | W      | 0000_0000h  | <a href="#">8.2.49/314</a> |
| CC4                  | Masked Parallel GPIO Pin Data Output Registers (SIU_MPGPDO17) | 32              | W      | 0000_0000h  | <a href="#">8.2.49/314</a> |
| CC8                  | Masked Parallel GPIO Pin Data Output Registers (SIU_MPGPDO18) | 32              | W      | 0000_0000h  | <a href="#">8.2.49/314</a> |
| CCC                  | Masked Parallel GPIO Pin Data Output Registers (SIU_MPGPDO19) | 32              | W      | 0000_0000h  | <a href="#">8.2.49/314</a> |
| CD0                  | Masked Parallel GPIO Pin Data Output Registers (SIU_MPGPDO20) | 32              | W      | 0000_0000h  | <a href="#">8.2.49/314</a> |
| CD4                  | Masked Parallel GPIO Pin Data Output Registers (SIU_MPGPDO21) | 32              | W      | 0000_0000h  | <a href="#">8.2.49/314</a> |
| CD8                  | Masked Parallel GPIO Pin Data Output Registers (SIU_MPGPDO22) | 32              | W      | 0000_0000h  | <a href="#">8.2.49/314</a> |
| CDC                  | Masked Parallel GPIO Pin Data Output Registers (SIU_MPGPDO23) | 32              | W      | 0000_0000h  | <a href="#">8.2.49/314</a> |
| CE0                  | Masked Parallel GPIO Pin Data Output Registers (SIU_MPGPDO24) | 32              | W      | 0000_0000h  | <a href="#">8.2.49/314</a> |
| CE4                  | Masked Parallel GPIO Pin Data Output Registers (SIU_MPGPDO25) | 32              | W      | 0000_0000h  | <a href="#">8.2.49/314</a> |

Table continues on the next page...



## SIU memory map (continued)

| Address offset (hex) | Register name                                                         | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|-----------------------------------------------------------------------|-----------------|--------|-------------|----------------------------|
| CE8                  | Masked Parallel GPIO Pin Data Output Registers (SIU_MPGPDO26)         | 32              | W      | 0000_0000h  | <a href="#">8.2.49/314</a> |
| CEC                  | Masked Parallel GPIO Pin Data Output Registers (SIU_MPGPDO27)         | 32              | W      | 0000_0000h  | <a href="#">8.2.49/314</a> |
| CF0                  | Masked Parallel GPIO Pin Data Output Registers (SIU_MPGPDO28)         | 32              | W      | 0000_0000h  | <a href="#">8.2.49/314</a> |
| CF4                  | Masked Parallel GPIO Pin Data Output Registers (SIU_MPGPDO29)         | 32              | W      | 0000_0000h  | <a href="#">8.2.49/314</a> |
| CF8                  | Masked Parallel GPIO Pin Data Output Registers (SIU_MPGPDO30)         | 32              | W      | 0000_0000h  | <a href="#">8.2.49/314</a> |
| CFC                  | Masked Parallel GPIO Pin Data Output Registers (SIU_MPGPDO31)         | 32              | W      | 0000_0000h  | <a href="#">8.2.49/314</a> |
| D00                  | Mask-Output High Register (SIU_DSPIAH)                                | 32              | R/W    | 0000_0000h  | <a href="#">8.2.50/318</a> |
| D04                  | Mask-Output Low Register (SIU_DSPIAL)                                 | 32              | R/W    | 0000_0000h  | <a href="#">8.2.51/322</a> |
| D08                  | Mask-Output High Register (SIU_DSPIBH)                                | 32              | R/W    | 0000_0000h  | <a href="#">8.2.50/318</a> |
| D0C                  | Mask-Output Low Register (SIU_DSPIBL)                                 | 32              | R/W    | 0000_0000h  | <a href="#">8.2.51/322</a> |
| D10                  | Mask-Output High Register (SIU_DSPICH)                                | 32              | R/W    | 0000_0000h  | <a href="#">8.2.50/318</a> |
| D14                  | Mask-Output Low Register (SIU_DSPICL)                                 | 32              | R/W    | 0000_0000h  | <a href="#">8.2.51/322</a> |
| D18                  | Mask-Output High Register (SIU_DSPIDH)                                | 32              | R/W    | 0000_0000h  | <a href="#">8.2.50/318</a> |
| D1C                  | Mask-Output Low Register (SIU_DSPIDL)                                 | 32              | R/W    | 0000_0000h  | <a href="#">8.2.51/322</a> |
| D40                  | Serialized Output Signal Selection Register for DSPI_A (SIU_ETPUBA)   | 32              | R/W    | 0000_0000h  | <a href="#">8.2.52/327</a> |
| D44                  | Serialized Output Signal Selection Register for DSPI_A (SIU_EMIOA)    | 32              | R/W    | 0000_0000h  | <a href="#">8.2.53/331</a> |
| D48                  | Serialized Output Signal Selection Register for DSPI_A (SIU_DSPIAHLA) | 32              | R/W    | 0000_0000h  | <a href="#">8.2.54/334</a> |
| D50                  | Serialized Output Signal Selection Register for DSPI_B (SIU_ETPUAB)   | 32              | R/W    | 0000_0000h  | <a href="#">8.2.55/338</a> |
| D54                  | Serialized Output Signal Selection Register for DSPI_B (SIU_EMIOB)    | 32              | R/W    | 0000_0000h  | <a href="#">8.2.56/342</a> |
| D58                  | Serialized Output Signal Selection Register for DSPI_B (SIU_DSPIBHLB) | 32              | R/W    | 0000_0000h  | <a href="#">8.2.57/346</a> |
| D60                  | Serialized Output Signal Selection Register for DSPI_C (SIU_ETPUAC)   | 32              | R/W    | 0000_0000h  | <a href="#">8.2.58/350</a> |
| D64                  | Serialized Output Signal Selection Register for DSPI_C (SIU_EMIOB)    | 32              | R/W    | 0000_0000h  | <a href="#">8.2.59/353</a> |
| D68                  | Serialized Output Signal Selection Register for DSPI_C (SIU_DSPICHL)  | 32              | R/W    | 0000_0000h  | <a href="#">8.2.60/357</a> |
| D6C                  | Serialized Output Signal Selection Register for DSPI_C (SIU_ETPUBC)   | 32              | R/W    | 0000_0000h  | <a href="#">8.2.61/361</a> |
| D70                  | Serialized Output Signal Selection Register for DSPI_D (SIU_ETPUBD)   | 32              | R/W    | 0000_0000h  | <a href="#">8.2.62/365</a> |
| D74                  | Serialized Output Signal Selection Register for DSPI_D (SIU_EMIOB)    | 32              | R/W    | 0000_0000h  | <a href="#">8.2.63/367</a> |

Table continues on the next page...

## SIU memory map (continued)

| Address offset (hex) | Register name                                                         | Width (in bits) | Access | Reset value                 | Section/ page |
|----------------------|-----------------------------------------------------------------------|-----------------|--------|-----------------------------|---------------|
| D78                  | Serialized Output Signal Selection Register for DSPI_D (SIU_DSPIDHLD) | 32              | R/W    | 0000_0000h                  | 8.2.64/368    |
| E4B                  | GPIO Pin Data Input Register (SIU_GPDI75)                             | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E4C                  | GPIO Pin Data Input Register (SIU_GPDI76)                             | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E4D                  | GPIO Pin Data Input Register (SIU_GPDI77)                             | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E4E                  | GPIO Pin Data Input Register (SIU_GPDI78)                             | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E4F                  | GPIO Pin Data Input Register (SIU_GPDI79)                             | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E50                  | GPIO Pin Data Input Register (SIU_GPDI80)                             | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E51                  | GPIO Pin Data Input Register (SIU_GPDI81)                             | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E52                  | GPIO Pin Data Input Register (SIU_GPDI82)                             | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E53                  | GPIO Pin Data Input Register (SIU_GPDI83)                             | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E54                  | GPIO Pin Data Input Register (SIU_GPDI84)                             | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E55                  | GPIO Pin Data Input Register (SIU_GPDI85)                             | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E56                  | GPIO Pin Data Input Register (SIU_GPDI86)                             | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E57                  | GPIO Pin Data Input Register (SIU_GPDI87)                             | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E58                  | GPIO Pin Data Input Register (SIU_GPDI88)                             | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E59                  | GPIO Pin Data Input Register (SIU_GPDI89)                             | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E5A                  | GPIO Pin Data Input Register (SIU_GPDI90)                             | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E5B                  | GPIO Pin Data Input Register (SIU_GPDI91)                             | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E5C                  | GPIO Pin Data Input Register (SIU_GPDI92)                             | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E5D                  | GPIO Pin Data Input Register (SIU_GPDI93)                             | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E5E                  | GPIO Pin Data Input Register (SIU_GPDI94)                             | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E5F                  | GPIO Pin Data Input Register (SIU_GPDI95)                             | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E60                  | GPIO Pin Data Input Register (SIU_GPDI96)                             | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E61                  | GPIO Pin Data Input Register (SIU_GPDI97)                             | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E62                  | GPIO Pin Data Input Register (SIU_GPDI98)                             | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E63                  | GPIO Pin Data Input Register (SIU_GPDI99)                             | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E64                  | GPIO Pin Data Input Register (SIU_GPDI100)                            | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E65                  | GPIO Pin Data Input Register (SIU_GPDI101)                            | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E66                  | GPIO Pin Data Input Register (SIU_GPDI102)                            | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E67                  | GPIO Pin Data Input Register (SIU_GPDI103)                            | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E68                  | GPIO Pin Data Input Register (SIU_GPDI104)                            | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E69                  | GPIO Pin Data Input Register (SIU_GPDI105)                            | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E6A                  | GPIO Pin Data Input Register (SIU_GPDI106)                            | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E6B                  | GPIO Pin Data Input Register (SIU_GPDI107)                            | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E6C                  | GPIO Pin Data Input Register (SIU_GPDI108)                            | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E6D                  | GPIO Pin Data Input Register (SIU_GPDI109)                            | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E6E                  | GPIO Pin Data Input Register (SIU_GPDI110)                            | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E6F                  | GPIO Pin Data Input Register (SIU_GPDI111)                            | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |

Table continues on the next page...

## SIU memory map (continued)

| Address offset (hex) | Register name                              | Width (in bits) | Access | Reset value                 | Section/ page |
|----------------------|--------------------------------------------|-----------------|--------|-----------------------------|---------------|
| E70                  | GPIO Pin Data Input Register (SIU_GPDI112) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E71                  | GPIO Pin Data Input Register (SIU_GPDI113) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E72                  | GPIO Pin Data Input Register (SIU_GPDI114) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E73                  | GPIO Pin Data Input Register (SIU_GPDI115) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E74                  | GPIO Pin Data Input Register (SIU_GPDI116) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E75                  | GPIO Pin Data Input Register (SIU_GPDI117) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E76                  | GPIO Pin Data Input Register (SIU_GPDI118) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E77                  | GPIO Pin Data Input Register (SIU_GPDI119) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E78                  | GPIO Pin Data Input Register (SIU_GPDI120) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E79                  | GPIO Pin Data Input Register (SIU_GPDI121) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E7A                  | GPIO Pin Data Input Register (SIU_GPDI122) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E7B                  | GPIO Pin Data Input Register (SIU_GPDI123) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E7C                  | GPIO Pin Data Input Register (SIU_GPDI124) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E7D                  | GPIO Pin Data Input Register (SIU_GPDI125) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E7E                  | GPIO Pin Data Input Register (SIU_GPDI126) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E7F                  | GPIO Pin Data Input Register (SIU_GPDI127) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E80                  | GPIO Pin Data Input Register (SIU_GPDI128) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E81                  | GPIO Pin Data Input Register (SIU_GPDI129) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E82                  | GPIO Pin Data Input Register (SIU_GPDI130) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E83                  | GPIO Pin Data Input Register (SIU_GPDI131) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E84                  | GPIO Pin Data Input Register (SIU_GPDI132) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E85                  | GPIO Pin Data Input Register (SIU_GPDI133) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E86                  | GPIO Pin Data Input Register (SIU_GPDI134) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E87                  | GPIO Pin Data Input Register (SIU_GPDI135) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E88                  | GPIO Pin Data Input Register (SIU_GPDI136) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E89                  | GPIO Pin Data Input Register (SIU_GPDI137) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E8A                  | GPIO Pin Data Input Register (SIU_GPDI138) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E8B                  | GPIO Pin Data Input Register (SIU_GPDI139) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E8C                  | GPIO Pin Data Input Register (SIU_GPDI140) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E8D                  | GPIO Pin Data Input Register (SIU_GPDI141) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E8E                  | GPIO Pin Data Input Register (SIU_GPDI142) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E8F                  | GPIO Pin Data Input Register (SIU_GPDI143) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E90                  | GPIO Pin Data Input Register (SIU_GPDI144) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E91                  | GPIO Pin Data Input Register (SIU_GPDI145) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E92                  | GPIO Pin Data Input Register (SIU_GPDI146) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E93                  | GPIO Pin Data Input Register (SIU_GPDI147) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E94                  | GPIO Pin Data Input Register (SIU_GPDI148) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E95                  | GPIO Pin Data Input Register (SIU_GPDI149) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E96                  | GPIO Pin Data Input Register (SIU_GPDI150) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |

Table continues on the next page...

## SIU memory map (continued)

| Address offset (hex) | Register name                              | Width (in bits) | Access | Reset value                 | Section/ page |
|----------------------|--------------------------------------------|-----------------|--------|-----------------------------|---------------|
| E97                  | GPIO Pin Data Input Register (SIU_GPDI151) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E98                  | GPIO Pin Data Input Register (SIU_GPDI152) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E99                  | GPIO Pin Data Input Register (SIU_GPDI153) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E9A                  | GPIO Pin Data Input Register (SIU_GPDI154) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E9B                  | GPIO Pin Data Input Register (SIU_GPDI155) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E9C                  | GPIO Pin Data Input Register (SIU_GPDI156) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E9D                  | GPIO Pin Data Input Register (SIU_GPDI157) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E9E                  | GPIO Pin Data Input Register (SIU_GPDI158) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| E9F                  | GPIO Pin Data Input Register (SIU_GPDI159) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EA0                  | GPIO Pin Data Input Register (SIU_GPDI160) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EA1                  | GPIO Pin Data Input Register (SIU_GPDI161) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EA2                  | GPIO Pin Data Input Register (SIU_GPDI162) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EA3                  | GPIO Pin Data Input Register (SIU_GPDI163) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EA4                  | GPIO Pin Data Input Register (SIU_GPDI164) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EA5                  | GPIO Pin Data Input Register (SIU_GPDI165) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EA6                  | GPIO Pin Data Input Register (SIU_GPDI166) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EA7                  | GPIO Pin Data Input Register (SIU_GPDI167) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EA8                  | GPIO Pin Data Input Register (SIU_GPDI168) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EA9                  | GPIO Pin Data Input Register (SIU_GPDI169) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EAA                  | GPIO Pin Data Input Register (SIU_GPDI170) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EAB                  | GPIO Pin Data Input Register (SIU_GPDI171) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EAC                  | GPIO Pin Data Input Register (SIU_GPDI172) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EAD                  | GPIO Pin Data Input Register (SIU_GPDI173) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EAE                  | GPIO Pin Data Input Register (SIU_GPDI174) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EAF                  | GPIO Pin Data Input Register (SIU_GPDI175) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EB0                  | GPIO Pin Data Input Register (SIU_GPDI176) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EB1                  | GPIO Pin Data Input Register (SIU_GPDI177) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EB2                  | GPIO Pin Data Input Register (SIU_GPDI178) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EB3                  | GPIO Pin Data Input Register (SIU_GPDI179) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EB4                  | GPIO Pin Data Input Register (SIU_GPDI180) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EB5                  | GPIO Pin Data Input Register (SIU_GPDI181) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EB6                  | GPIO Pin Data Input Register (SIU_GPDI182) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EB7                  | GPIO Pin Data Input Register (SIU_GPDI183) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EB8                  | GPIO Pin Data Input Register (SIU_GPDI184) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EB9                  | GPIO Pin Data Input Register (SIU_GPDI185) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EBA                  | GPIO Pin Data Input Register (SIU_GPDI186) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EBB                  | GPIO Pin Data Input Register (SIU_GPDI187) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EBC                  | GPIO Pin Data Input Register (SIU_GPDI188) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EBD                  | GPIO Pin Data Input Register (SIU_GPDI189) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |

Table continues on the next page...

## SIU memory map (continued)

| Address offset (hex) | Register name                              | Width (in bits) | Access | Reset value                 | Section/ page |
|----------------------|--------------------------------------------|-----------------|--------|-----------------------------|---------------|
| EBE                  | GPIO Pin Data Input Register (SIU_GPDI190) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EBF                  | GPIO Pin Data Input Register (SIU_GPDI191) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EC0                  | GPIO Pin Data Input Register (SIU_GPDI192) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EC1                  | GPIO Pin Data Input Register (SIU_GPDI193) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EC2                  | GPIO Pin Data Input Register (SIU_GPDI194) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EC3                  | GPIO Pin Data Input Register (SIU_GPDI195) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EC4                  | GPIO Pin Data Input Register (SIU_GPDI196) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EC5                  | GPIO Pin Data Input Register (SIU_GPDI197) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EC6                  | GPIO Pin Data Input Register (SIU_GPDI198) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EC7                  | GPIO Pin Data Input Register (SIU_GPDI199) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EC8                  | GPIO Pin Data Input Register (SIU_GPDI200) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EC9                  | GPIO Pin Data Input Register (SIU_GPDI201) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| ECA                  | GPIO Pin Data Input Register (SIU_GPDI202) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| ECB                  | GPIO Pin Data Input Register (SIU_GPDI203) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| ECC                  | GPIO Pin Data Input Register (SIU_GPDI204) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| ECD                  | GPIO Pin Data Input Register (SIU_GPDI205) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| ECE                  | GPIO Pin Data Input Register (SIU_GPDI206) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| ECF                  | GPIO Pin Data Input Register (SIU_GPDI207) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| ED0                  | GPIO Pin Data Input Register (SIU_GPDI208) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| ED1                  | GPIO Pin Data Input Register (SIU_GPDI209) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| ED2                  | GPIO Pin Data Input Register (SIU_GPDI210) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| ED3                  | GPIO Pin Data Input Register (SIU_GPDI211) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| ED4                  | GPIO Pin Data Input Register (SIU_GPDI212) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| ED5                  | GPIO Pin Data Input Register (SIU_GPDI213) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| ED6                  | GPIO Pin Data Input Register (SIU_GPDI214) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| ED7                  | GPIO Pin Data Input Register (SIU_GPDI215) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| ED8                  | GPIO Pin Data Input Register (SIU_GPDI216) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| ED9                  | GPIO Pin Data Input Register (SIU_GPDI217) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EDA                  | GPIO Pin Data Input Register (SIU_GPDI218) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EDB                  | GPIO Pin Data Input Register (SIU_GPDI219) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EDC                  | GPIO Pin Data Input Register (SIU_GPDI220) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EDD                  | GPIO Pin Data Input Register (SIU_GPDI221) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EDE                  | GPIO Pin Data Input Register (SIU_GPDI222) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EDF                  | GPIO Pin Data Input Register (SIU_GPDI223) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EE0                  | GPIO Pin Data Input Register (SIU_GPDI224) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EE1                  | GPIO Pin Data Input Register (SIU_GPDI225) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EE2                  | GPIO Pin Data Input Register (SIU_GPDI226) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EE3                  | GPIO Pin Data Input Register (SIU_GPDI227) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |
| EE4                  | GPIO Pin Data Input Register (SIU_GPDI228) | 8               | R      | <a href="#">See section</a> | 8.2.65/372    |

Table continues on the next page...



## SIU memory map (continued)

| Address offset (hex) | Register name                              | Width (in bits) | Access | Reset value                 | Section/ page              |
|----------------------|--------------------------------------------|-----------------|--------|-----------------------------|----------------------------|
| EE5                  | GPIO Pin Data Input Register (SIU_GPDI229) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| EE6                  | GPIO Pin Data Input Register (SIU_GPDI230) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| EE7                  | GPIO Pin Data Input Register (SIU_GPDI231) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| EE8                  | GPIO Pin Data Input Register (SIU_GPDI232) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| EE9                  | GPIO Pin Data Input Register (SIU_GPDI233) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| EEA                  | GPIO Pin Data Input Register (SIU_GPDI234) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| EEB                  | GPIO Pin Data Input Register (SIU_GPDI235) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| EEC                  | GPIO Pin Data Input Register (SIU_GPDI236) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| EED                  | GPIO Pin Data Input Register (SIU_GPDI237) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| EEE                  | GPIO Pin Data Input Register (SIU_GPDI238) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| EEF                  | GPIO Pin Data Input Register (SIU_GPDI239) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| EF0                  | GPIO Pin Data Input Register (SIU_GPDI240) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| EF1                  | GPIO Pin Data Input Register (SIU_GPDI241) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| EF2                  | GPIO Pin Data Input Register (SIU_GPDI242) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| EF3                  | GPIO Pin Data Input Register (SIU_GPDI243) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| EF4                  | GPIO Pin Data Input Register (SIU_GPDI244) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| EF5                  | GPIO Pin Data Input Register (SIU_GPDI245) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| EF6                  | GPIO Pin Data Input Register (SIU_GPDI246) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| EF7                  | GPIO Pin Data Input Register (SIU_GPDI247) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| EF8                  | GPIO Pin Data Input Register (SIU_GPDI248) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| EF9                  | GPIO Pin Data Input Register (SIU_GPDI249) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| EFA                  | GPIO Pin Data Input Register (SIU_GPDI250) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| EFB                  | GPIO Pin Data Input Register (SIU_GPDI251) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| EFC                  | GPIO Pin Data Input Register (SIU_GPDI252) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| EFD                  | GPIO Pin Data Input Register (SIU_GPDI253) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| EFE                  | GPIO Pin Data Input Register (SIU_GPDI254) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| EFF                  | GPIO Pin Data Input Register (SIU_GPDI255) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F00                  | GPIO Pin Data Input Register (SIU_GPDI256) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F01                  | GPIO Pin Data Input Register (SIU_GPDI257) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F02                  | GPIO Pin Data Input Register (SIU_GPDI258) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F03                  | GPIO Pin Data Input Register (SIU_GPDI259) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F04                  | GPIO Pin Data Input Register (SIU_GPDI260) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F05                  | GPIO Pin Data Input Register (SIU_GPDI261) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F06                  | GPIO Pin Data Input Register (SIU_GPDI262) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F07                  | GPIO Pin Data Input Register (SIU_GPDI263) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F08                  | GPIO Pin Data Input Register (SIU_GPDI264) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F09                  | GPIO Pin Data Input Register (SIU_GPDI265) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F0A                  | GPIO Pin Data Input Register (SIU_GPDI266) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F0B                  | GPIO Pin Data Input Register (SIU_GPDI267) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |

Table continues on the next page...

## SIU memory map (continued)

| Address offset (hex) | Register name                              | Width (in bits) | Access | Reset value                 | Section/ page              |
|----------------------|--------------------------------------------|-----------------|--------|-----------------------------|----------------------------|
| F0C                  | GPIO Pin Data Input Register (SIU_GPDI268) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F0D                  | GPIO Pin Data Input Register (SIU_GPDI269) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F0E                  | GPIO Pin Data Input Register (SIU_GPDI270) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F0F                  | GPIO Pin Data Input Register (SIU_GPDI271) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F10                  | GPIO Pin Data Input Register (SIU_GPDI272) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F11                  | GPIO Pin Data Input Register (SIU_GPDI273) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F12                  | GPIO Pin Data Input Register (SIU_GPDI274) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F13                  | GPIO Pin Data Input Register (SIU_GPDI275) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F14                  | GPIO Pin Data Input Register (SIU_GPDI276) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F15                  | GPIO Pin Data Input Register (SIU_GPDI277) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F16                  | GPIO Pin Data Input Register (SIU_GPDI278) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F17                  | GPIO Pin Data Input Register (SIU_GPDI279) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F18                  | GPIO Pin Data Input Register (SIU_GPDI280) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F19                  | GPIO Pin Data Input Register (SIU_GPDI281) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F1A                  | GPIO Pin Data Input Register (SIU_GPDI282) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F1B                  | GPIO Pin Data Input Register (SIU_GPDI283) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F1C                  | GPIO Pin Data Input Register (SIU_GPDI284) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F1D                  | GPIO Pin Data Input Register (SIU_GPDI285) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F1E                  | GPIO Pin Data Input Register (SIU_GPDI286) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F1F                  | GPIO Pin Data Input Register (SIU_GPDI287) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F20                  | GPIO Pin Data Input Register (SIU_GPDI288) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F21                  | GPIO Pin Data Input Register (SIU_GPDI289) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F22                  | GPIO Pin Data Input Register (SIU_GPDI290) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F23                  | GPIO Pin Data Input Register (SIU_GPDI291) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F24                  | GPIO Pin Data Input Register (SIU_GPDI292) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F25                  | GPIO Pin Data Input Register (SIU_GPDI293) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F26                  | GPIO Pin Data Input Register (SIU_GPDI294) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F27                  | GPIO Pin Data Input Register (SIU_GPDI295) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F28                  | GPIO Pin Data Input Register (SIU_GPDI296) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F29                  | GPIO Pin Data Input Register (SIU_GPDI297) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F2A                  | GPIO Pin Data Input Register (SIU_GPDI298) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F2B                  | GPIO Pin Data Input Register (SIU_GPDI299) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F2C                  | GPIO Pin Data Input Register (SIU_GPDI300) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F2D                  | GPIO Pin Data Input Register (SIU_GPDI301) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F2E                  | GPIO Pin Data Input Register (SIU_GPDI302) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F2F                  | GPIO Pin Data Input Register (SIU_GPDI303) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F30                  | GPIO Pin Data Input Register (SIU_GPDI304) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F31                  | GPIO Pin Data Input Register (SIU_GPDI305) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F32                  | GPIO Pin Data Input Register (SIU_GPDI306) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |

Table continues on the next page...

## SIU memory map (continued)

| Address offset (hex) | Register name                              | Width (in bits) | Access | Reset value                 | Section/ page              |
|----------------------|--------------------------------------------|-----------------|--------|-----------------------------|----------------------------|
| F33                  | GPIO Pin Data Input Register (SIU_GPDI307) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F34                  | GPIO Pin Data Input Register (SIU_GPDI308) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F35                  | GPIO Pin Data Input Register (SIU_GPDI309) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F36                  | GPIO Pin Data Input Register (SIU_GPDI310) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F37                  | GPIO Pin Data Input Register (SIU_GPDI311) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F38                  | GPIO Pin Data Input Register (SIU_GPDI312) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F39                  | GPIO Pin Data Input Register (SIU_GPDI313) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F3A                  | GPIO Pin Data Input Register (SIU_GPDI314) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F3B                  | GPIO Pin Data Input Register (SIU_GPDI315) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F3C                  | GPIO Pin Data Input Register (SIU_GPDI316) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F3D                  | GPIO Pin Data Input Register (SIU_GPDI317) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F3E                  | GPIO Pin Data Input Register (SIU_GPDI318) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F3F                  | GPIO Pin Data Input Register (SIU_GPDI319) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F40                  | GPIO Pin Data Input Register (SIU_GPDI320) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F41                  | GPIO Pin Data Input Register (SIU_GPDI321) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F42                  | GPIO Pin Data Input Register (SIU_GPDI322) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F43                  | GPIO Pin Data Input Register (SIU_GPDI323) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F44                  | GPIO Pin Data Input Register (SIU_GPDI324) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F45                  | GPIO Pin Data Input Register (SIU_GPDI325) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F46                  | GPIO Pin Data Input Register (SIU_GPDI326) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F47                  | GPIO Pin Data Input Register (SIU_GPDI327) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F48                  | GPIO Pin Data Input Register (SIU_GPDI328) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F49                  | GPIO Pin Data Input Register (SIU_GPDI329) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F4A                  | GPIO Pin Data Input Register (SIU_GPDI330) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F4B                  | GPIO Pin Data Input Register (SIU_GPDI331) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F4C                  | GPIO Pin Data Input Register (SIU_GPDI332) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F4D                  | GPIO Pin Data Input Register (SIU_GPDI333) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F4E                  | GPIO Pin Data Input Register (SIU_GPDI334) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F4F                  | GPIO Pin Data Input Register (SIU_GPDI335) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F50                  | GPIO Pin Data Input Register (SIU_GPDI336) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F51                  | GPIO Pin Data Input Register (SIU_GPDI337) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F52                  | GPIO Pin Data Input Register (SIU_GPDI338) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F53                  | GPIO Pin Data Input Register (SIU_GPDI339) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F54                  | GPIO Pin Data Input Register (SIU_GPDI340) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F55                  | GPIO Pin Data Input Register (SIU_GPDI341) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F56                  | GPIO Pin Data Input Register (SIU_GPDI342) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F57                  | GPIO Pin Data Input Register (SIU_GPDI343) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F58                  | GPIO Pin Data Input Register (SIU_GPDI344) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F59                  | GPIO Pin Data Input Register (SIU_GPDI345) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |

Table continues on the next page...



## SIU memory map (continued)

| Address offset (hex) | Register name                              | Width (in bits) | Access | Reset value                 | Section/ page              |
|----------------------|--------------------------------------------|-----------------|--------|-----------------------------|----------------------------|
| F5A                  | GPIO Pin Data Input Register (SIU_GPDI346) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F5B                  | GPIO Pin Data Input Register (SIU_GPDI347) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F5C                  | GPIO Pin Data Input Register (SIU_GPDI348) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F5D                  | GPIO Pin Data Input Register (SIU_GPDI349) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F5E                  | GPIO Pin Data Input Register (SIU_GPDI350) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F5F                  | GPIO Pin Data Input Register (SIU_GPDI351) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F60                  | GPIO Pin Data Input Register (SIU_GPDI352) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F61                  | GPIO Pin Data Input Register (SIU_GPDI353) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F62                  | GPIO Pin Data Input Register (SIU_GPDI354) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F63                  | GPIO Pin Data Input Register (SIU_GPDI355) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F64                  | GPIO Pin Data Input Register (SIU_GPDI356) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F65                  | GPIO Pin Data Input Register (SIU_GPDI357) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F66                  | GPIO Pin Data Input Register (SIU_GPDI358) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F67                  | GPIO Pin Data Input Register (SIU_GPDI359) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F68                  | GPIO Pin Data Input Register (SIU_GPDI360) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F69                  | GPIO Pin Data Input Register (SIU_GPDI361) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F6A                  | GPIO Pin Data Input Register (SIU_GPDI362) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F6B                  | GPIO Pin Data Input Register (SIU_GPDI363) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F6C                  | GPIO Pin Data Input Register (SIU_GPDI364) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F6D                  | GPIO Pin Data Input Register (SIU_GPDI365) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F6E                  | GPIO Pin Data Input Register (SIU_GPDI366) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F6F                  | GPIO Pin Data Input Register (SIU_GPDI367) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F70                  | GPIO Pin Data Input Register (SIU_GPDI368) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F71                  | GPIO Pin Data Input Register (SIU_GPDI369) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F72                  | GPIO Pin Data Input Register (SIU_GPDI370) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F73                  | GPIO Pin Data Input Register (SIU_GPDI371) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F74                  | GPIO Pin Data Input Register (SIU_GPDI372) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F75                  | GPIO Pin Data Input Register (SIU_GPDI373) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F76                  | GPIO Pin Data Input Register (SIU_GPDI374) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F77                  | GPIO Pin Data Input Register (SIU_GPDI375) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F78                  | GPIO Pin Data Input Register (SIU_GPDI376) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F79                  | GPIO Pin Data Input Register (SIU_GPDI377) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F7A                  | GPIO Pin Data Input Register (SIU_GPDI378) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F7B                  | GPIO Pin Data Input Register (SIU_GPDI379) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F7C                  | GPIO Pin Data Input Register (SIU_GPDI380) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F7D                  | GPIO Pin Data Input Register (SIU_GPDI381) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F7E                  | GPIO Pin Data Input Register (SIU_GPDI382) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F7F                  | GPIO Pin Data Input Register (SIU_GPDI383) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F80                  | GPIO Pin Data Input Register (SIU_GPDI384) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |

Table continues on the next page...

## SIU memory map (continued)

| Address offset (hex) | Register name                              | Width (in bits) | Access | Reset value                 | Section/ page              |
|----------------------|--------------------------------------------|-----------------|--------|-----------------------------|----------------------------|
| F81                  | GPIO Pin Data Input Register (SIU_GPDI385) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F82                  | GPIO Pin Data Input Register (SIU_GPDI386) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F83                  | GPIO Pin Data Input Register (SIU_GPDI387) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F84                  | GPIO Pin Data Input Register (SIU_GPDI388) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F85                  | GPIO Pin Data Input Register (SIU_GPDI389) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F86                  | GPIO Pin Data Input Register (SIU_GPDI390) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F87                  | GPIO Pin Data Input Register (SIU_GPDI391) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F88                  | GPIO Pin Data Input Register (SIU_GPDI392) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F89                  | GPIO Pin Data Input Register (SIU_GPDI393) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F8A                  | GPIO Pin Data Input Register (SIU_GPDI394) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F8B                  | GPIO Pin Data Input Register (SIU_GPDI395) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F8C                  | GPIO Pin Data Input Register (SIU_GPDI396) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F8D                  | GPIO Pin Data Input Register (SIU_GPDI397) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F8E                  | GPIO Pin Data Input Register (SIU_GPDI398) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F8F                  | GPIO Pin Data Input Register (SIU_GPDI399) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F90                  | GPIO Pin Data Input Register (SIU_GPDI400) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F91                  | GPIO Pin Data Input Register (SIU_GPDI401) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F92                  | GPIO Pin Data Input Register (SIU_GPDI402) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F93                  | GPIO Pin Data Input Register (SIU_GPDI403) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F94                  | GPIO Pin Data Input Register (SIU_GPDI404) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F95                  | GPIO Pin Data Input Register (SIU_GPDI405) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F96                  | GPIO Pin Data Input Register (SIU_GPDI406) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F97                  | GPIO Pin Data Input Register (SIU_GPDI407) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F98                  | GPIO Pin Data Input Register (SIU_GPDI408) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F99                  | GPIO Pin Data Input Register (SIU_GPDI409) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F9A                  | GPIO Pin Data Input Register (SIU_GPDI410) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F9B                  | GPIO Pin Data Input Register (SIU_GPDI411) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F9C                  | GPIO Pin Data Input Register (SIU_GPDI412) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F9D                  | GPIO Pin Data Input Register (SIU_GPDI413) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F9E                  | GPIO Pin Data Input Register (SIU_GPDI414) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| F9F                  | GPIO Pin Data Input Register (SIU_GPDI415) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FA0                  | GPIO Pin Data Input Register (SIU_GPDI416) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FA1                  | GPIO Pin Data Input Register (SIU_GPDI417) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FA2                  | GPIO Pin Data Input Register (SIU_GPDI418) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FA3                  | GPIO Pin Data Input Register (SIU_GPDI419) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FA4                  | GPIO Pin Data Input Register (SIU_GPDI420) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FA5                  | GPIO Pin Data Input Register (SIU_GPDI421) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FA6                  | GPIO Pin Data Input Register (SIU_GPDI422) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FA7                  | GPIO Pin Data Input Register (SIU_GPDI423) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |

Table continues on the next page...

## SIU memory map (continued)

| Address offset (hex) | Register name                              | Width (in bits) | Access | Reset value                 | Section/ page              |
|----------------------|--------------------------------------------|-----------------|--------|-----------------------------|----------------------------|
| FA8                  | GPIO Pin Data Input Register (SIU_GPDI424) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FA9                  | GPIO Pin Data Input Register (SIU_GPDI425) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FAA                  | GPIO Pin Data Input Register (SIU_GPDI426) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FAB                  | GPIO Pin Data Input Register (SIU_GPDI427) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FAC                  | GPIO Pin Data Input Register (SIU_GPDI428) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FAD                  | GPIO Pin Data Input Register (SIU_GPDI429) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FAE                  | GPIO Pin Data Input Register (SIU_GPDI430) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FAF                  | GPIO Pin Data Input Register (SIU_GPDI431) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FB0                  | GPIO Pin Data Input Register (SIU_GPDI432) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FB1                  | GPIO Pin Data Input Register (SIU_GPDI433) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FB2                  | GPIO Pin Data Input Register (SIU_GPDI434) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FB3                  | GPIO Pin Data Input Register (SIU_GPDI435) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FB4                  | GPIO Pin Data Input Register (SIU_GPDI436) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FB5                  | GPIO Pin Data Input Register (SIU_GPDI437) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FB6                  | GPIO Pin Data Input Register (SIU_GPDI438) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FB7                  | GPIO Pin Data Input Register (SIU_GPDI439) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FB8                  | GPIO Pin Data Input Register (SIU_GPDI440) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FB9                  | GPIO Pin Data Input Register (SIU_GPDI441) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FBA                  | GPIO Pin Data Input Register (SIU_GPDI442) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FBB                  | GPIO Pin Data Input Register (SIU_GPDI443) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FBC                  | GPIO Pin Data Input Register (SIU_GPDI444) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FBD                  | GPIO Pin Data Input Register (SIU_GPDI445) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FBE                  | GPIO Pin Data Input Register (SIU_GPDI446) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FBF                  | GPIO Pin Data Input Register (SIU_GPDI447) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FC0                  | GPIO Pin Data Input Register (SIU_GPDI448) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FC1                  | GPIO Pin Data Input Register (SIU_GPDI449) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FC2                  | GPIO Pin Data Input Register (SIU_GPDI450) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FC3                  | GPIO Pin Data Input Register (SIU_GPDI451) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FC4                  | GPIO Pin Data Input Register (SIU_GPDI452) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FC5                  | GPIO Pin Data Input Register (SIU_GPDI453) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FC6                  | GPIO Pin Data Input Register (SIU_GPDI454) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FC7                  | GPIO Pin Data Input Register (SIU_GPDI455) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FC8                  | GPIO Pin Data Input Register (SIU_GPDI456) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FC9                  | GPIO Pin Data Input Register (SIU_GPDI457) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FCA                  | GPIO Pin Data Input Register (SIU_GPDI458) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FCB                  | GPIO Pin Data Input Register (SIU_GPDI459) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FCC                  | GPIO Pin Data Input Register (SIU_GPDI460) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FCD                  | GPIO Pin Data Input Register (SIU_GPDI461) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FCE                  | GPIO Pin Data Input Register (SIU_GPDI462) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |

Table continues on the next page...

## SIU memory map (continued)

| Address offset (hex) | Register name                              | Width (in bits) | Access | Reset value                 | Section/ page              |
|----------------------|--------------------------------------------|-----------------|--------|-----------------------------|----------------------------|
| FCF                  | GPIO Pin Data Input Register (SIU_GPDI463) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FD0                  | GPIO Pin Data Input Register (SIU_GPDI464) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FD1                  | GPIO Pin Data Input Register (SIU_GPDI465) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FD2                  | GPIO Pin Data Input Register (SIU_GPDI466) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FD3                  | GPIO Pin Data Input Register (SIU_GPDI467) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FD4                  | GPIO Pin Data Input Register (SIU_GPDI468) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FD5                  | GPIO Pin Data Input Register (SIU_GPDI469) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FD6                  | GPIO Pin Data Input Register (SIU_GPDI470) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FD7                  | GPIO Pin Data Input Register (SIU_GPDI471) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FD8                  | GPIO Pin Data Input Register (SIU_GPDI472) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FD9                  | GPIO Pin Data Input Register (SIU_GPDI473) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FDA                  | GPIO Pin Data Input Register (SIU_GPDI474) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FDB                  | GPIO Pin Data Input Register (SIU_GPDI475) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FDC                  | GPIO Pin Data Input Register (SIU_GPDI476) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FDD                  | GPIO Pin Data Input Register (SIU_GPDI477) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FDE                  | GPIO Pin Data Input Register (SIU_GPDI478) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FDF                  | GPIO Pin Data Input Register (SIU_GPDI479) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FE0                  | GPIO Pin Data Input Register (SIU_GPDI480) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FE1                  | GPIO Pin Data Input Register (SIU_GPDI481) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FE2                  | GPIO Pin Data Input Register (SIU_GPDI482) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FE3                  | GPIO Pin Data Input Register (SIU_GPDI483) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FE4                  | GPIO Pin Data Input Register (SIU_GPDI484) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FE5                  | GPIO Pin Data Input Register (SIU_GPDI485) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FE6                  | GPIO Pin Data Input Register (SIU_GPDI486) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FE7                  | GPIO Pin Data Input Register (SIU_GPDI487) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FE8                  | GPIO Pin Data Input Register (SIU_GPDI488) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FE9                  | GPIO Pin Data Input Register (SIU_GPDI489) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FEA                  | GPIO Pin Data Input Register (SIU_GPDI490) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| FEB                  | GPIO Pin Data Input Register (SIU_GPDI491) | 8               | R      | <a href="#">See section</a> | <a href="#">8.2.65/372</a> |
| 1200                 | Input Multiplexing Register0 (SIU_IMUX0)   | 32              | R/W    | 0000_0000h                  | <a href="#">8.2.66/373</a> |
| 1204                 | Input Multiplexing Register1 (SIU_IMUX1)   | 32              | R/W    | 0000_0000h                  | <a href="#">8.2.67/375</a> |
| 1208                 | Input Multiplexing Register2 (SIU_IMUX2)   | 32              | R/W    | 0000_0000h                  | <a href="#">8.2.68/376</a> |
| 120C                 | Input Multiplexing Register3 (SIU_IMUX3)   | 32              | R/W    | 0000_0000h                  | <a href="#">8.2.69/378</a> |
| 1210                 | Input Multiplexing Register4 (SIU_IMUX4)   | 32              | R/W    | 0000_0000h                  | <a href="#">8.2.70/379</a> |
| 1214                 | Input Multiplexing Register5 (SIU_IMUX5)   | 32              | R/W    | 0000_0000h                  | <a href="#">8.2.71/381</a> |
| 121C                 | Input Multiplexing Register7 (SIU_IMUX7)   | 32              | R/W    | 0000_0000h                  | <a href="#">8.2.72/383</a> |
| 1228                 | Input Multiplexing Register10 (SIU_IMUX10) | 32              | R/W    | 0000_0000h                  | <a href="#">8.2.73/385</a> |
| 1230                 | Input Multiplexing Register12 (SIU_IMUX12) | 32              | R/W    | 0000_0000h                  | <a href="#">8.2.74/387</a> |

## 8.2.1 MCU Identification Register (SIU\_MIDR)

The SIU\_MIDR contains the part number and mask number revisions specific to the device. The part number is a read-only field that is mask programmed with the part number of the device on which the SIU resides. It is intended to help factory test and user code which are sensitive to part changes. The part number is changed if a new module is added to the device or a memory size is changed, for example. It is not changed for bug fixes or process changes.

The mask number is a read-only field that is mask programmed with the specific mask revision level of the device. It is intended to help factory test and user code which is sensitive to mask changes.

Address: 0h base + 4h offset = 4h

|       |         |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |     |    |    |    |    |    |                |    |    |    |                |    |    |    |    |    |
|-------|---------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|-----|----|----|----|----|----|----------------|----|----|----|----------------|----|----|----|----|----|
| Bit   | 0       | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16  | 17 | 18 | 19 | 20 | 21 | 22             | 23 | 24 | 25 | 26             | 27 | 28 | 29 | 30 | 31 |
| R     | PARTNUM |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | PKG |    |    | 0  |    |    | MASKNUM_ MAJOR |    |    |    | MASKNUM_ MINOR |    |    |    |    |    |
| W     |         |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |     |    |    |    |    |    |                |    |    |    |                |    |    |    |    |    |
| Reset | 0       | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1  | 1  | 0  | 1  | 1  | 1  | *   | *  | *  | *  | 0  | 0  | 0              | 0  | *  | *  | *              | *  | *  | *  | *  | *  |

\* Notes:

- MASKNUM\_MINOR field: 1N45H = 0001b, 2N45H = 0010b, 3N45H = 0000b
- MASKNUM\_MAJOR field: 1N45H = 0000b, 2N45H = 0000b, 3N45H = 0001b
- PKG field: 416 = 0110b, 516 = 1110b, 552 = 1100b

### SIU\_MIDR field descriptions

| Field                   | Description                                                                                                                                                                  |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>PARTNUM         | MCU Part Number. Read-only, mask programmed part number of the MCU. MPC5777 reads 0x5777.                                                                                    |
| 16–19<br>PKG            | Package settings. Read-only, package information.<br>0110: 416-pin package.<br>1110: 516-pin package.<br>1100: 552-pin package.<br>All other settings reserved.              |
| 20–23<br>Reserved       | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                      |
| 24–27<br>MASKNUM_ MAJOR | Major revision of MCU Mask. Read-only, mask programmed mask number of the MCU. Reads 0x0 for the initial mask set of the device, and changes sequentially for each mask set. |
| 28–31<br>MASKNUM_ MINOR | Minor revision MCU Mask. Read-only, mask programmed mask number of the MCU. Reads 0x0 for the initial mask set of the device, and changes sequentially for each mask set.    |

8.2.2 Reset Status Register (SIU\_RSR)

The Reset Status Register contains the sources of the most recent reset as well as a number of configuration bits defined by the Miscellaneous DCF record, which the SSCM reads from flash memory during reset. Alternatively, if the LEG bit of the Miscellaneous DCF record is set to 1, then the reset controller assumes that a backwards compatible legacy mode must be used. In this case, the values of the configuration bits are latched from package pins at the end of the reset cycle instead of coming from the DCF record.

The reset values of status bits are applied at POR. The reset value of the configuration bits (WKPCFG, XOSCHIGH, ABR, BOOTCFG) are loaded from the Miscellaneous DCF record or from external pins.

Address: 0h base + Ch offset = Ch

| Bit   | 0    | 1   | 2  | 3  | 4  | 5  | 6  | 7  | 8     | 9  | 10     | 11     | 12     | 13   | 14   | 15  |
|-------|------|-----|----|----|----|----|----|----|-------|----|--------|--------|--------|------|------|-----|
| R     | PORS | ERS | 0  |    |    |    |    |    | CPURS | 0  | FOSURS | FCCURS | STCURS | SSRS | SERF |     |
| W     |      |     |    |    |    |    |    |    |       |    |        |        |        |      |      | w1c |
| Reset | 1*   | 0*  | 0* | 0* | 0* | 0* | 0* | 0* | 0*    | 0* | 0*     | 0*     | 0*     | 0*   | 0*   | 0*  |

| Bit   | 16     | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24   | 25 | 26 | 27       | 28  | 29      | 30 | 31  |
|-------|--------|----|----|----|----|----|----|----|------|----|----|----------|-----|---------|----|-----|
| R     | WKPCFG |    |    |    | 0  |    |    |    | XOSC |    | 0  | XOSCHIGH | ABR | BOOTCFG |    | RGF |
| W     |        |    |    |    |    |    |    |    |      |    |    |          |     |         |    | w1c |
| Reset | *      | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0*   | 0* | 0* | *        | *   | *       | *  | 0*  |

\* Notes:

- BOOTCFG field: The reset value of this configuration field is loaded from the Miscellaneous DCF record or from external pins.
- ABR field: The reset value of this configuration field is loaded from the Miscellaneous DCF record or from external pins.
- XOSCHIGH field: The reset value of this configuration field is loaded from the Miscellaneous DCF record or from external pins.
- WKPCFG field: The reset value of this configuration field is loaded from the Miscellaneous DCF record or from external pins.

### SIU\_RSR field descriptions

| Field           | Description                                                                                                                                                                                                                                                                                                                                        |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>PORS       | Power-on reset status.<br><br>0 Another reset source has been acknowledged by the reset controller since the last assertion of the power-on reset input.<br>1 The power-on reset input to the reset controller was asserted and no other reset source was acknowledged since that assertion of the power-on reset input, except an external reset. |
| 1<br>ERS        | External reset status.<br><br>0 The last reset source acknowledged by the reset controller was not a valid assertion of the RESET pin.<br>1 The last reset source acknowledged by the reset controller was a valid assertion of the RESET pin.                                                                                                     |
| 2–8<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                            |
| 9<br>CPURS      | Simultaneous Core 0 and Core 1 reset status.<br><br>0 The last reset source acknowledged by the reset controller was not a simultaneous reset attempt of both cores.<br>1 The last reset source acknowledged by the reset controller was a simultaneous reset attempt of both cores.                                                               |
| 10<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                            |

Table continues on the next page...

## SIU\_RSR field descriptions (continued)

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 11<br>FOSURS      | FCCU Output Supervision Unit (FOSU) reset status.<br>0 The last reset source acknowledged by the reset controller was not a reset from the FOSU.<br>1 The last reset source acknowledged by the reset controller was a reset from the FOSU.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 12<br>FCCURS      | Failure Collection and Control Unit (FCCU) reset status.<br>0 The last reset source acknowledged by the reset controller was not a reset from the FCCU.<br>1 The last reset source acknowledged by the reset controller was a reset from the FCCU.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 13<br>STCURS      | Self Test Control Unit (STCU) reset status.<br>0 The last reset source acknowledged by the reset controller was not a reset from the STCU.<br>1 The last reset source acknowledged by the reset controller was a reset from the STCU.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 14<br>SSRS        | Software system reset status.<br>0 The last reset source acknowledged by the reset controller was not a software system reset.<br>1 The last reset source acknowledged by the reset controller was a software system reset.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 15<br>SERF        | Software external reset flag.<br>0 The software external reset input to the reset controller was not asserted, or this bit has been cleared by writing a 1 to it.<br>1 The software external reset input to the reset controller was asserted while this bit was zero.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 16<br>WKPCFG      | Weak pull-up configuration. The value of this bit is loaded from bit 12 of the Miscellaneous DCF record by the SSCM during reset. If the MSB bit of the DCF record (Legacy Bit) is set to '1', then the value of this bit is latched from the WKPCFG pin upon negation of reset, instead of being loaded from the DCF record.<br>0 Weak pulldown is the default setting.<br>1 Weak pullup is the default setting.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 17–23<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 24<br>XOSC        | External oscillator status. This is a set-once bit that is set after an internal counter (clocked by XOSC) finishes counting to 512 times the XOSC_CNT field of the Miscellaneous DCF record. When set, this flag indicates that the XOSC clock is stable and can be used as clock source for system and peripherals.<br>0 XOSC clock not stable yet.<br>1 XOSC clock is stable.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 25–26<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 27<br>XOSCHIGH    | XOSC at High Setting. The value of this bit is loaded from bit 21 of the Miscellaneous DCF record by the SSCM during reset. However, if the MSB of the DCF record (Legacy bit) is set to 1, then the value of this bit is latched from the PLLCFG[2] pin upon negation of reset instead of being loaded from the DCF record.<br><b>NOTE:</b> "Low" is the setting commonly used for crystals at 8 MHz, "medium" is commonly used for crystals greater than 8 MHz to 20 MHz, and "high" is commonly used for crystals greater than 20 MHz to 40 MHz. However, the user must characterize carefully to determine the best $g_m$ setting for the intended application because crystal load capacitance, board layout, and other factors affect the $g_m$ value that is needed. See the Data Sheet for additional information.<br>0 Crystal in the range of low to medium setting.<br>1 Crystal is at high setting. |
| 28<br>ABR         | Auto baud rate configuration. The value of this bit is loaded from bit 13 of the Miscellaneous DCF record by the SSCM during reset. If the MSB bit of the DCF record (Legacy Bit) is set to '1', then the value of this                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

Table continues on the next page...



## SIU\_RSR field descriptions (continued)

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                  | bit is determined by the value of the EVTO pin upon negation of reset as follows. If no debug tool is attached to the device and EVTO is held at low logic level during reset, the ABR bit reset state is logic one. Otherwise it is zero.<br><br>0 Auto Baud Rate Disabled.<br>1 Auto Baud Rate Enabled.                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 29–30<br>BOOTCFG | Boot configuration. BOOTCFG[0:1] is used by the Boot Assist Module (BAM) to configure the boot mode. If the component life cycle is "in field", then the value of BOOTCFG is fixed to 0b00 (internal boot). For all other life cycles, the value of BOOTCFG is loaded from bits 14-15 of the Miscellaneous DCF record by the SSCM during reset. Alternatively, if the MSB bit of the Miscellaneous DCF record (Legacy Bit) is set to '1', then the value of this field is latched from the BOOTCFG pins upon negation of reset, instead of being loaded from the DCF record.<br><br>00 Internal boot mode.<br>01 Serial boot mode.<br>10 Non-muxed EBI boot mode (16-bit data, non-muxed address bus).<br>11 Muxed EBI boot mode (16-bit data, muxed address bus). |
| 31<br>RGF        | Reset glitch flag. Set by the reset controller when a glitch is detected on the RESET pin. This bit is cleared by the assertion of the power-on reset or a write of one to the RGF bit.<br><br>0 No glitch has been detected on the RESET pin.<br>1 A glitch has been detected on the RESET pin.                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

## 8.2.3 System Reset Control Register (SIU\_SRCR)

The system reset control register configures whether a software system reset or a software external reset is generated. A software system reset uses an internal system reset. A software external reset asserts  $\overline{\text{RSTOUT}}$ .

Address: 0h base + 10h offset = 10h

|       |     |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-----|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0   | 1   | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0   |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     | SSR | SER |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0   | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| Bit   | 16  | 17  | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |     |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |     |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0   | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

## SIU\_SRCR field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>SSR         | Software system reset. The software system reset is processed as a synchronous reset. Except for a software external reset, the bit automatically clears if any other reset source asserts.<br><br><b>NOTE:</b> The SSR bit always reads as zero. A write of zero to this bit has no effect.<br><br>0 No software system reset.<br>1 Generate a software internal system reset.                                                                                                                                                                                                                                                                                                                              |
| 1<br>SER         | Software external reset. Used to generate a Software External Reset. Writing 1 to this bit asserts RSTOUT for 2480 clock cycles, but the internal reset is not asserted. The bit is automatically cleared when the software external reset completes or any other reset source is asserted. After a software external reset has been initiated, RSTOUT negates if this bit is cleared before the 2480 clock cycles period expires.<br><br><b>NOTE:</b> Write 1 to the SER bit to generate a software external reset. A write of 0 to this bit has no effect. When reset completes, the SER bit is cleared to 0.<br><br>0 Do not generate a software external reset.<br>1 Generate a software external reset. |
| 2–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |

## 8.2.4 External IRQ Status Register (SIU\_EISR)

The external interrupt status register is used to record edge-triggered events on the  $\overline{\text{IRQ}}[0]\text{--}\overline{\text{IRQ}}[15]$  inputs to the SIU and record the critical interrupts NMI and SWT. When an edge-detect enable bit is set in the SIU\_IREER or SIU\_IFEER registers for an IRQ and an IRQ edge-event is detected, the flag bit is set in the SIU\_EISR. The flag bits are cleared by writing 1 to the bit. Writing 0 has no effect.

The flag bit is set regardless of the state of the DMA or interrupt request enable bit in SIU\_DIRER. The flag bit remains set until cleared by software or through the servicing of a DMA or interrupt request.

Address: 0h base + 14h offset = 14h

| Bit   | 0    | 1    | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|-------|------|------|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| R     | NMI0 | NMI1 | 0 |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     | w1c  | w1c  |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0    | 0    | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |       |       |       |       |       |       |      |      |      |      |      |      |      |      |      |      |
|-------|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| Bit   | 16    | 17    | 18    | 19    | 20    | 21    | 22   | 23   | 24   | 25   | 26   | 27   | 28   | 29   | 30   | 31   |
| R     | EIF15 | EIF14 | EIF13 | EIF12 | EIF11 | EIF10 | EIF9 | EIF8 | EIF7 | EIF6 | EIF5 | EIF4 | EIF3 | EIF2 | EIF1 | EIF0 |
| W     | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c  | w1c  | w1c  | w1c  | w1c  | w1c  | w1c  | w1c  | w1c  | w1c  |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |

## SIU\_EISR field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                       |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>NMI0        | Core0 non-maskable interrupt flag. This bit is set when a NMI interrupt occurs on the NMI input pin of Core0.<br><br>0 No NMI event has occurred on the NMI input of Core0.<br>1 An NMI event has occurred on the NMI input of Core0.                                                                                                                             |
| 1<br>NMI1        | Core1 non-maskable interrupt flag. This bit is set when a NMI interrupt occurs on the NMI input pin of Core1.<br><br>0 No NMI event has occurred on the NMI input of Core1.<br>1 An NMI event has occurred on the NMI input of Core1.                                                                                                                             |
| 2–15<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                           |
| 16<br>EIF15      | External interrupt request flag 15. This bit is set when an edge-triggered event occurs on the corresponding $\overline{\text{IRQ}}[15]$ input. Cleared by writing 1.<br><br>0 No edge triggered event has occurred on the corresponding $\overline{\text{IRQ}}$ pin.<br>1 An edge triggered event has occurred on the corresponding $\overline{\text{IRQ}}$ pin. |
| 17<br>EIF14      | External interrupt request flag 14. This bit is set when an edge-triggered event occurs on the corresponding $\overline{\text{IRQ}}[14]$ input. Cleared by writing 1.<br><br>0 No edge triggered event has occurred on the corresponding $\overline{\text{IRQ}}$ pin.<br>1 An edge triggered event has occurred on the corresponding $\overline{\text{IRQ}}$ pin. |
| 18<br>EIF13      | External interrupt request flag 13. This bit is set when an edge-triggered event occurs on the corresponding $\overline{\text{IRQ}}[13]$ input. Cleared by writing 1.<br><br>0 No edge triggered event has occurred on the corresponding $\overline{\text{IRQ}}$ pin.<br>1 An edge triggered event has occurred on the corresponding $\overline{\text{IRQ}}$ pin. |
| 19<br>EIF12      | External interrupt request flag 12. This bit is set when an edge-triggered event occurs on the corresponding $\overline{\text{IRQ}}[12]$ input. Cleared by writing 1.<br><br>0 No edge triggered event has occurred on the corresponding $\overline{\text{IRQ}}$ pin.<br>1 An edge triggered event has occurred on the corresponding $\overline{\text{IRQ}}$ pin. |
| 20<br>EIF11      | External interrupt request flag 11. This bit is set when an edge-triggered event occurs on the corresponding $\overline{\text{IRQ}}[11]$ input. Cleared by writing 1.<br><br>0 No edge triggered event has occurred on the corresponding $\overline{\text{IRQ}}$ pin.<br>1 An edge triggered event has occurred on the corresponding $\overline{\text{IRQ}}$ pin. |
| 21<br>EIF10      | External interrupt request flag 10. This bit is set when an edge-triggered event occurs on the corresponding $\overline{\text{IRQ}}[10]$ input. Cleared by writing 1.                                                                                                                                                                                             |

Table continues on the next page...

## SIU\_EISR field descriptions (continued)

| Field      | Description                                                                                                                                                                                                                                                                                                                                                     |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|            | 0 No edge triggered event has occurred on the corresponding $\overline{\text{IRQ}}$ pin.<br>1 An edge triggered event has occurred on the corresponding $\overline{\text{IRQ}}$ pin.                                                                                                                                                                            |
| 22<br>EIF9 | External interrupt request flag 9. This bit is set when an edge-triggered event occurs on the corresponding $\overline{\text{IRQ}}[9]$ input. Cleared by writing 1.<br><br>0 No edge triggered event has occurred on the corresponding $\overline{\text{IRQ}}$ pin.<br>1 An edge triggered event has occurred on the corresponding $\overline{\text{IRQ}}$ pin. |
| 23<br>EIF8 | External interrupt request flag 8. This bit is set when an edge-triggered event occurs on the corresponding $\overline{\text{IRQ}}[8]$ input. Cleared by writing 1.<br><br>0 No edge triggered event has occurred on the corresponding $\overline{\text{IRQ}}$ pin.<br>1 An edge triggered event has occurred on the corresponding $\overline{\text{IRQ}}$ pin. |
| 24<br>EIF7 | External interrupt request flag 7. This bit is set when an edge-triggered event occurs on the corresponding $\overline{\text{IRQ}}[7]$ input. Cleared by writing 1.<br><br>0 No edge triggered event has occurred on the corresponding $\overline{\text{IRQ}}$ pin.<br>1 An edge triggered event has occurred on the corresponding $\overline{\text{IRQ}}$ pin. |
| 25<br>EIF6 | External interrupt request flag 6. This bit is set when an edge-triggered event occurs on the corresponding $\overline{\text{IRQ}}[6]$ input. Cleared by writing 1.<br><br>0 No edge triggered event has occurred on the corresponding $\overline{\text{IRQ}}$ pin.<br>1 An edge triggered event has occurred on the corresponding $\overline{\text{IRQ}}$ pin. |
| 26<br>EIF5 | External interrupt request flag 5. This bit is set when an edge-triggered event occurs on the corresponding $\overline{\text{IRQ}}[5]$ input. Cleared by writing 1.<br><br>0 No edge triggered event has occurred on the corresponding $\overline{\text{IRQ}}$ pin.<br>1 An edge triggered event has occurred on the corresponding $\overline{\text{IRQ}}$ pin. |
| 27<br>EIF4 | External interrupt request flag 4. This bit is set when an edge-triggered event occurs on the corresponding $\overline{\text{IRQ}}[4]$ input. Cleared by writing 1.<br><br>0 No edge triggered event has occurred on the corresponding $\overline{\text{IRQ}}$ pin.<br>1 An edge triggered event has occurred on the corresponding $\overline{\text{IRQ}}$ pin. |
| 28<br>EIF3 | External interrupt request flag 3. This bit is set when an edge-triggered event occurs on the corresponding $\overline{\text{IRQ}}[3]$ input. Cleared by writing 1.<br><br>0 No edge triggered event has occurred on the corresponding $\overline{\text{IRQ}}$ pin.<br>1 An edge triggered event has occurred on the corresponding $\overline{\text{IRQ}}$ pin. |
| 29<br>EIF2 | External interrupt request flag 2. This bit is set when an edge-triggered event occurs on the corresponding $\overline{\text{IRQ}}[2]$ input. Cleared by writing 1.<br><br>0 No edge triggered event has occurred on the corresponding $\overline{\text{IRQ}}$ pin.<br>1 An edge triggered event has occurred on the corresponding $\overline{\text{IRQ}}$ pin. |
| 30<br>EIF1 | External interrupt request flag 1. This bit is set when an edge-triggered event occurs on the corresponding $\overline{\text{IRQ}}[1]$ input. Cleared by writing 1.<br><br>0 No edge triggered event has occurred on the corresponding $\overline{\text{IRQ}}$ pin.<br>1 An edge triggered event has occurred on the corresponding $\overline{\text{IRQ}}$ pin. |
| 31<br>EIF0 | External interrupt request flag 0. This bit is set when an edge-triggered event occurs on the corresponding $\overline{\text{IRQ}}[0]$ input. Cleared by writing 1.                                                                                                                                                                                             |

Table continues on the next page...

## SIU\_EISR field descriptions (continued)

| Field | Description                                                                            |
|-------|----------------------------------------------------------------------------------------|
| 0     | No edge triggered event has occurred on the corresponding $\overline{\text{IRQ}}$ pin. |
| 1     | An edge triggered event has occurred on the corresponding $\overline{\text{IRQ}}$ pin. |

## 8.2.5 DMA/Interrupt Request Enable Register (SIU\_DIRER)

The SIU\_DIRER enables external signals on selected pins to generate DMA or interrupt requests, and the SWT modules to generate interrupt requests. The EIRE[0:3] bits enable requests to either the DMA or INTC modules, based on the value of the DIRSn bits in the SIU\_DIRSR register. The EIRE[4:15] bits enable requests only to the INTC module. The NMISELn bits enable requests to either the critical or non-maskable interrupt inputs of the cores. For a pin to generate an interrupt, the appropriate bit in either SIU\_IREER or SIU\_IFEER registers must additionally be set.

Address: 0h base + 18h offset = 18h

|       |         |         |        |        |        |        |       |       |         |         |       |       |       |       |       |       |
|-------|---------|---------|--------|--------|--------|--------|-------|-------|---------|---------|-------|-------|-------|-------|-------|-------|
| Bit   | 0       | 1       | 2      | 3      | 4      | 5      | 6     | 7     | 8       | 9       | 10    | 11    | 12    | 13    | 14    | 15    |
| R     | NMISEL8 | NMISEL7 | 0      |        |        |        |       |       | NMISEL0 | NMISEL1 | 0     |       |       |       |       |       |
| W     |         |         |        |        |        |        |       |       |         |         |       |       |       |       |       |       |
| Reset | 0       | 0       | 0      | 0      | 0      | 0      | 0     | 0     | 0       | 0       | 0     | 0     | 0     | 0     | 0     | 0     |
| Bit   | 16      | 17      | 18     | 19     | 20     | 21     | 22    | 23    | 24      | 25      | 26    | 27    | 28    | 29    | 30    | 31    |
| R     | EIRE15  | EIRE14  | EIRE13 | EIRE12 | EIRE11 | EIRE10 | EIRE9 | EIRE8 | EIRE7   | EIRE6   | EIRE5 | EIRE4 | EIRE3 | EIRE2 | EIRE1 | EIRE0 |
| W     |         |         |        |        |        |        |       |       |         |         |       |       |       |       |       |       |
| Reset | 0       | 0       | 0      | 0      | 0      | 0      | 0     | 0     | 0       | 0       | 0     | 0     | 0     | 0     | 0     | 0     |

## SIU\_DIRER field descriptions

| Field        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>NMISEL8 | Core0 non-maskable/critical interrupt selection (NMI comes from external pin). The NMI external pin can be routed to Core0 either as critical interrupt or as non-maskable interrupt (NMI). The NMISEL8 bit selects which of them is driven by the NMI pin. This bit is write-once.<br><br><b>NOTE:</b> NMISEL8 is a write-once bit.<br><br>0 Core0 NMI interrupt receives the signal from the NMI pin (IVOR1 core exception).<br>1 Core0 critical interrupt receives the signal from the NMI pin (IVOR0 core exception). |
| 1<br>NMISEL7 | Core1 non-maskable/critical interrupt selection (NMI comes from external pin). The NMI external pin can be routed to Core1 either as critical interrupt or as non-maskable interrupt (NMI). The NMISEL7 bit selects which of them is driven by the NMI pin. This bit is write-once.                                                                                                                                                                                                                                       |

Table continues on the next page...

## SIU\_DIRER field descriptions (continued)

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | <p><b>NOTE:</b> NMISEL7 is a write-once bit.</p> <p>0 Core1 NMI interrupt receives the signal from the NMI pin (IVOR1 core exception).<br/> 1 Core1 critical interrupt receives the signal from the NMI pin (IVOR0 core exception).</p>                                                                                                                                                                                                                                                                                                     |
| 2–7<br>Reserved   | <p>This field is reserved.<br/> This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 8<br>NMISEL0      | <p>Core0 non-maskable/critical interrupt selection (NMI comes from SWT_A). The Software Watchdog Timer A (SWT_A) interrupt output can be routed to Core0 either as critical interrupt or as non-maskable interrupt (NMI). The NMISEL0 bit selects which of them is driven by SWT_A. This bit is write-once.</p> <p><b>NOTE:</b> NMISEL0 is a write-once bit.</p> <p>0 Core0 NMI interrupt receives the signal from SWT_A (IVOR1 core exception).<br/> 1 Core0 critical interrupt receives the signal from SWT_A (IVOR0 core exception).</p> |
| 9<br>NMISEL1      | <p>Core1 non-maskable/critical interrupt selection (NMI comes from SWT_B). The Software Watchdog Timer B (SWT_B) interrupt output can be routed to Core1 either as critical interrupt or as non-maskable interrupt (NMI). The NMISEL1 bit selects which of them is driven by SWT_B. This bit is write-once.</p> <p><b>NOTE:</b> NMISEL1 is a write-once bit.</p> <p>0 Core1 NMI interrupt receives the signal from SWT_B (IVOR1 core exception).<br/> 1 Core1 critical interrupt receives the signal from SWT_B (IVOR0 core exception).</p> |
| 10–15<br>Reserved | <p>This field is reserved.<br/> This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 16<br>EIRE15      | <p>Interrupt request enable 15. This bit enables the assertion of an interrupt request when an edge-triggered event occurs on the <math>\overline{\text{IRQ}}[15]</math> pin.</p> <p>0 Interrupt request is disabled.<br/> 1 Interrupt request is enabled.</p>                                                                                                                                                                                                                                                                              |
| 17<br>EIRE14      | <p>Interrupt request enable 14. This bit enables the assertion of an interrupt request when an edge-triggered event occurs on the <math>\overline{\text{IRQ}}[14]</math> pin.</p> <p>0 Interrupt request is disabled.<br/> 1 Interrupt request is enabled.</p>                                                                                                                                                                                                                                                                              |
| 18<br>EIRE13      | <p>Interrupt request enable 13. This bit enables the assertion of an interrupt request when an edge-triggered event occurs on the <math>\overline{\text{IRQ}}[13]</math> pin.</p> <p>0 Interrupt request is disabled.<br/> 1 Interrupt request is enabled.</p>                                                                                                                                                                                                                                                                              |
| 19<br>EIRE12      | <p>Interrupt request enable 12. This bit enables the assertion of an interrupt request when an edge-triggered event occurs on the <math>\overline{\text{IRQ}}[12]</math> pin.</p> <p>0 Interrupt request is disabled.<br/> 1 Interrupt request is enabled.</p>                                                                                                                                                                                                                                                                              |
| 20<br>EIRE11      | <p>Interrupt request enable 11. This bit enables the assertion of an interrupt request when an edge-triggered event occurs on the <math>\overline{\text{IRQ}}[11]</math> pin.</p> <p>0 Interrupt request is disabled.<br/> 1 Interrupt request is enabled.</p>                                                                                                                                                                                                                                                                              |
| 21<br>EIRE10      | <p>Interrupt request enable 10. This bit enables the assertion of an interrupt request when an edge-triggered event occurs on the <math>\overline{\text{IRQ}}[10]</math> pin.</p>                                                                                                                                                                                                                                                                                                                                                           |

Table continues on the next page...

## SIU\_DIRER field descriptions (continued)

| Field       | Description                                                                                                                                                                                                                                                                                                                                                      |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|             | 0 Interrupt request is disabled.<br>1 Interrupt request is enabled.                                                                                                                                                                                                                                                                                              |
| 22<br>EIRE9 | Interrupt request enable 9. This bit enables the assertion of an interrupt request when an edge-triggered event occurs on the $\overline{\text{IRQ}}[9]$ pin.<br><br>0 Interrupt request is disabled.<br>1 Interrupt request is enabled.                                                                                                                         |
| 23<br>EIRE8 | Interrupt request enable 8. This bit enables the assertion of an interrupt request when an edge-triggered event occurs on the $\overline{\text{IRQ}}[8]$ pin.<br><br>0 Interrupt request is disabled.<br>1 Interrupt request is enabled.                                                                                                                         |
| 24<br>EIRE7 | Interrupt request enable 7. This bit enables the assertion of an interrupt request when an edge-triggered event occurs on the $\overline{\text{IRQ}}[7]$ pin.<br><br>0 Interrupt request is disabled.<br>1 Interrupt request is enabled.                                                                                                                         |
| 25<br>EIRE6 | Interrupt request enable 6. This bit enables the assertion of an interrupt request when an edge-triggered event occurs on the $\overline{\text{IRQ}}[6]$ pin.<br><br>0 Interrupt request is disabled.<br>1 Interrupt request is enabled.                                                                                                                         |
| 26<br>EIRE5 | Interrupt request enable 5. This bit enables the assertion of an interrupt request when an edge-triggered event occurs on the $\overline{\text{IRQ}}[5]$ pin.<br><br>0 Interrupt request is disabled.<br>1 Interrupt request is enabled.                                                                                                                         |
| 27<br>EIRE4 | Interrupt request enable 4. This bit enables the assertion of an interrupt request when an edge-triggered event occurs on the $\overline{\text{IRQ}}[4]$ pin.<br><br>0 Interrupt request is disabled.<br>1 Interrupt request is enabled.                                                                                                                         |
| 28<br>EIRE3 | DMA/interrupt request enable 3. This bit enables the assertion of a DMA request or an interrupt request when an edge-triggered event occurs on the $\overline{\text{IRQ}}[3]$ pin.<br><br><b>NOTE:</b> EIRE[0:3] can optionally enable DMA requests instead of IRQs.<br><br>0 DMA and interrupt requests are disabled.<br>1 DMA or interrupt request is enabled. |
| 29<br>EIRE2 | DMA/interrupt request enable 2. This bit enables the assertion of a DMA request or an interrupt request when an edge-triggered event occurs on the $\overline{\text{IRQ}}[2]$ pin.<br><br><b>NOTE:</b> EIRE[0:3] can optionally enable DMA requests instead of IRQs.<br><br>0 DMA and interrupt requests are disabled.<br>1 DMA or interrupt request is enabled. |
| 30<br>EIRE1 | DMA/interrupt request enable 1. This bit enables the assertion of a DMA request or an interrupt request when an edge-triggered event occurs on the $\overline{\text{IRQ}}[1]$ pin.<br><br><b>NOTE:</b> EIRE[0:3] can optionally enable DMA requests instead of IRQs.                                                                                             |

*Table continues on the next page...*

## SIU\_DIRER field descriptions (continued)

| Field       | Description                                                                                                                                                                                                                                                                                                                                                      |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|             | 0 DMA and interrupt requests are disabled.<br>1 DMA or interrupt request is enabled.                                                                                                                                                                                                                                                                             |
| 31<br>EIRE0 | DMA/interrupt request enable 0. This bit enables the assertion of a DMA request or an interrupt request when an edge-triggered event occurs on the $\overline{\text{IRQ}}[0]$ pin.<br><br><b>NOTE:</b> EIRE[0:3] can optionally enable DMA requests instead of IRQs.<br><br>0 DMA and interrupt requests are disabled.<br>1 DMA or interrupt request is enabled. |

## 8.2.6 DMA/Interrupt Request Select Register (SIU\_DIRSR)

The SIU\_DIRSR selects between DMA and interrupt requests for events on the  $\overline{\text{IRQ}}[0]$ - $\overline{\text{IRQ}}[3]$ . If the corresponding bits are set in SIU\_EISR and SIU\_DIRER, then the DMA/Interrupt Request Select bit (DIRS $n$ ) determines whether a DMA or interrupt request is asserted.

Address: 0h base + 1Ch offset = 1Ch

|       |    |    |    |    |    |    |    |    |    |    |    |    |       |       |       |       |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|-------|-------|-------|-------|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12    | 13    | 14    | 15    |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |       |       |       |       |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |       |       |       |       |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0     | 0     | 0     | 0     |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28    | 29    | 30    | 31    |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    | DIRS3 | DIRS2 | DIRS1 | DIRS0 |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |       |       |       |       |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0     | 0     | 0     | 0     |

## SIU\_DIRSR field descriptions

| Field            | Description                                                                                                                                                                                                                                   |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–27<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                       |
| 28<br>DIRS3      | DMA/interrupt select 3. Selects between a DMA or external interrupt request when an edge-triggered event occurs on $\overline{\text{IRQ}}[3]$ pin.<br><br>0 Interrupt request is selected.<br>1 DMA request to eDMA_A channel 51 is selected. |

Table continues on the next page...



**SIU\_DIRSR field descriptions (continued)**

| <b>Field</b> | <b>Description</b>                                                                                                                                                                                                                                                    |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 29<br>DIRS2  | DMA/interrupt select 2. Selects between a DMA or external interrupt request when an edge-triggered event occurs on the corresponding $\overline{\text{IRQ}}[2]$ pin.<br><br>0    Interrupt request is selected.<br>1    DMA request to eDMA_A channel 50 is selected. |
| 30<br>DIRS1  | DMA/interrupt select 1. Selects between a DMA or external interrupt request when an edge-triggered event occurs on the corresponding $\overline{\text{IRQ}}[1]$ pin.<br><br>0    Interrupt request is selected.<br>1    DMA request to eDMA_A channel 49 is selected. |
| 31<br>DIRS0  | DMA/interrupt select 0. Selects between a DMA or external interrupt request when an edge-triggered event occurs on the corresponding $\overline{\text{IRQ}}[0]$ pin.<br><br>0    Interrupt request is selected.<br>1    DMA request to eDMA_A channel 48 is selected. |

## 8.2.7 Overrun Status Register (SIU\_OSR)

The SIU\_OSR flag bits indicate that an overrun has occurred. An overrun is generated if an event on an  $\overline{\text{IRQ}}$  pin occurs while the corresponding flag bit is set in SIU\_EISR. The overrun flag bit is set regardless of the state of the corresponding DMA/Interrupt Request Enable bit in SIU\_DIRER or the corresponding Overrun Request Enable bit in SIU\_ORER. The numbering of the Overrun Flag bits correspond to the numbering of the flag bits in SIU\_EISR. The Overrun Flag bit remains set until cleared by software. Each Overrun Flag bit is cleared by writing 1. Writing 0 has no effect.

Address: 0h base + 20h offset = 20h

| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

| Bit   | 16    | 17    | 18    | 19    | 20    | 21    | 22   | 23   | 24   | 25   | 26   | 27   | 28   | 29   | 30   | 31   |
|-------|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| R     | OVF15 | OVF14 | OVF13 | OVF12 | OVF11 | OVF10 | OVF9 | OVF8 | OVF7 | OVF6 | OVF5 | OVF4 | OVF3 | OVF2 | OVF1 | OVF0 |
| W     | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c  | w1c  | w1c  | w1c  | w1c  | w1c  | w1c  | w1c  | w1c  | w1c  |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |

### SIU\_OSR field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                    |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                        |
| 16<br>OVF15      | Overrun flag 15. This bit is set when an interrupt overrun occurs on $\overline{\text{IRQ}}[15]$ . Bit 31 (OVF0) is the overrun flag for $\overline{\text{IRQ}}[0]$ ; bit 16 (OVF15) is overrun flag for $\overline{\text{IRQ}}[15]$ .<br><br>0 No overrun occurred.<br>1 An overrun occurred. |
| 17<br>OVF14      | Overrun flag 14. This bit is set when an interrupt overrun occurs on $\overline{\text{IRQ}}[14]$ . Bit 31 (OVF0) is the overrun flag for $\overline{\text{IRQ}}[0]$ ; bit 16 (OVF15) is overrun flag for $\overline{\text{IRQ}}[15]$ .                                                         |

Table continues on the next page...

**SIU\_OSR field descriptions (continued)**

| Field       | Description                                                                                                                                                                                                                                                                                    |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|             | 0 No overrun occurred.<br>1 An overrun occurred.                                                                                                                                                                                                                                               |
| 18<br>OVF13 | Overrun flag 13. This bit is set when an interrupt overrun occurs on $\overline{\text{IRQ}}[13]$ . Bit 31 (OVF0) is the overrun flag for $\overline{\text{IRQ}}[0]$ ; bit 16 (OVF15) is overrun flag for $\overline{\text{IRQ}}[15]$ .<br><br>0 No overrun occurred.<br>1 An overrun occurred. |
| 19<br>OVF12 | Overrun flag 12. This bit is set when an interrupt overrun occurs on $\overline{\text{IRQ}}[12]$ . Bit 31 (OVF0) is the overrun flag for $\overline{\text{IRQ}}[0]$ ; bit 16 (OVF15) is overrun flag for $\overline{\text{IRQ}}[15]$ .<br><br>0 No overrun occurred.<br>1 An overrun occurred. |
| 20<br>OVF11 | Overrun flag 11. This bit is set when an interrupt overrun occurs on $\overline{\text{IRQ}}[11]$ . Bit 31 (OVF0) is the overrun flag for $\overline{\text{IRQ}}[0]$ ; bit 16 (OVF15) is overrun flag for $\overline{\text{IRQ}}[15]$ .<br><br>0 No overrun occurred.<br>1 An overrun occurred. |
| 21<br>OVF10 | Overrun flag 10. This bit is set when an interrupt overrun occurs on $\overline{\text{IRQ}}[10]$ . Bit 31 (OVF0) is the overrun flag for $\overline{\text{IRQ}}[0]$ ; bit 16 (OVF15) is overrun flag for $\overline{\text{IRQ}}[15]$ .<br><br>0 No overrun occurred.<br>1 An overrun occurred. |
| 22<br>OVF9  | Overrun flag 9. This bit is set when an interrupt overrun occurs on $\overline{\text{IRQ}}[9]$ . Bit 31 (OVF0) is the overrun flag for $\overline{\text{IRQ}}[0]$ ; bit 16 (OVF15) is overrun flag for $\overline{\text{IRQ}}[15]$ .<br><br>0 No overrun occurred.<br>1 An overrun occurred.   |
| 23<br>OVF8  | Overrun flag 8. This bit is set when an interrupt overrun occurs on $\overline{\text{IRQ}}[8]$ . Bit 31 (OVF0) is the overrun flag for $\overline{\text{IRQ}}[0]$ ; bit 16 (OVF15) is overrun flag for $\overline{\text{IRQ}}[15]$ .<br><br>0 No overrun occurred.<br>1 An overrun occurred.   |
| 24<br>OVF7  | Overrun flag 7. This bit is set when an interrupt overrun occurs on $\overline{\text{IRQ}}[7]$ . Bit 31 (OVF0) is the overrun flag for $\overline{\text{IRQ}}[0]$ ; bit 16 (OVF15) is overrun flag for $\overline{\text{IRQ}}[15]$ .<br><br>0 No overrun occurred.<br>1 An overrun occurred.   |
| 25<br>OVF6  | Overrun flag 6. This bit is set when an interrupt overrun occurs on $\overline{\text{IRQ}}[6]$ . Bit 31 (OVF0) is the overrun flag for $\overline{\text{IRQ}}[0]$ ; bit 16 (OVF15) is overrun flag for $\overline{\text{IRQ}}[15]$ .<br><br>0 No overrun occurred.<br>1 An overrun occurred.   |
| 26<br>OVF5  | Overrun flag 5. This bit is set when an interrupt overrun occurs on $\overline{\text{IRQ}}[5]$ . Bit 31 (OVF0) is the overrun flag for $\overline{\text{IRQ}}[0]$ ; bit 16 (OVF15) is overrun flag for $\overline{\text{IRQ}}[15]$ .<br><br>0 No overrun occurred.<br>1 An overrun occurred.   |
| 27<br>OVF4  | Overrun flag 4. This bit is set when an interrupt overrun occurs on $\overline{\text{IRQ}}[4]$ . Bit 31 (OVF0) is the overrun flag for $\overline{\text{IRQ}}[0]$ ; bit 16 (OVF15) is overrun flag for $\overline{\text{IRQ}}[15]$ .                                                           |

*Table continues on the next page...*

## SIU\_OSR field descriptions (continued)

| Field      | Description                                                                                                                                                                                                                                                                                        |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|            | 0 No overrun occurred.<br>1 An overrun occurred.                                                                                                                                                                                                                                                   |
| 28<br>OVF3 | Overrun flag 3. This bit is set when a DMA or interrupt overrun occurs on $\overline{\text{IRQ}}[3]$ . Bit 31 (OVF0) is the overrun flag for $\overline{\text{IRQ}}[0]$ ; bit 16 (OVF15) is overrun flag for $\overline{\text{IRQ}}[15]$ .<br><br>0 No overrun occurred.<br>1 An overrun occurred. |
| 29<br>OVF2 | Overrun flag 2. This bit is set when a DMA or interrupt overrun occurs on $\overline{\text{IRQ}}[2]$ . Bit 31 (OVF0) is the overrun flag for $\overline{\text{IRQ}}[0]$ ; bit 16 (OVF15) is overrun flag for $\overline{\text{IRQ}}[15]$ .<br><br>0 No overrun occurred.<br>1 An overrun occurred. |
| 30<br>OVF1 | Overrun flag 1. This bit is set when a DMA or interrupt overrun occurs on $\overline{\text{IRQ}}[1]$ . Bit 31 (OVF0) is the overrun flag for $\overline{\text{IRQ}}[0]$ ; bit 16 (OVF15) is overrun flag for $\overline{\text{IRQ}}[15]$ .<br><br>0 No overrun occurred.<br>1 An overrun occurred. |
| 31<br>OVF0 | Overrun flag 0. This bit is set when a DMA or interrupt overrun occurs on $\overline{\text{IRQ}}[0]$ . Bit 31 (OVF0) is the overrun flag for $\overline{\text{IRQ}}[0]$ ; bit 16 (OVF15) is overrun flag for $\overline{\text{IRQ}}[15]$ .<br><br>0 No overrun occurred.<br>1 An overrun occurred. |

## 8.2.8 Overrun Request Enable Register (SIU\_ORER)

The SIU\_ORER contains bits to enable an overrun if the corresponding flag bit is set in the SIU\_OSR. If the overrun request enable bit and the flag bit are set, the single combined overrun request from the SIU to the interrupt controller is asserted.

Address: 0h base + 24h offset = 24h

|       |       |       |       |       |       |       |      |      |      |      |      |      |      |      |      |      |
|-------|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| Bit   | 0     | 1     | 2     | 3     | 4     | 5     | 6    | 7    | 8    | 9    | 10   | 11   | 12   | 13   | 14   | 15   |
| R     | 0     |       |       |       |       |       |      |      |      |      |      |      |      |      |      |      |
| W     |       |       |       |       |       |       |      |      |      |      |      |      |      |      |      |      |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |
| Bit   | 16    | 17    | 18    | 19    | 20    | 21    | 22   | 23   | 24   | 25   | 26   | 27   | 28   | 29   | 30   | 31   |
| R     | ORE15 | ORE14 | ORE13 | ORE12 | ORE11 | ORE10 | ORE9 | ORE8 | ORE7 | ORE6 | ORE5 | ORE4 | ORE3 | ORE2 | ORE1 | ORE0 |
| W     | ORE15 | ORE14 | ORE13 | ORE12 | ORE11 | ORE10 | ORE9 | ORE8 | ORE7 | ORE6 | ORE5 | ORE4 | ORE3 | ORE2 | ORE1 | ORE0 |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |

## SIU\_ORE field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                        |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                            |
| 16<br>ORE15      | Overflow request enable 15. Enables the overflow request when an overflow occurs in the $\overline{\text{IRQ}}[15]$ pin. Bit 31 (ORE0) is the enable overflow flag for $\overline{\text{IRQ}}[0]$ ; bit 16 (ORE15) is the overflow flag for $\overline{\text{IRQ}}[15]$ .<br><br>0 Overflow request is disabled.<br>1 Overflow request is enabled. |
| 17<br>ORE14      | Overflow request enable 14. Enables the overflow request when an overflow occurs in the $\overline{\text{IRQ}}[14]$ pin. Bit 31 (ORE0) is the enable overflow flag for $\overline{\text{IRQ}}[0]$ ; bit 16 (ORE15) is the overflow flag for $\overline{\text{IRQ}}[15]$ .<br><br>0 Overflow request is disabled.<br>1 Overflow request is enabled. |
| 18<br>ORE13      | Overflow request enable 13. Enables the overflow request when an overflow occurs in the $\overline{\text{IRQ}}[13]$ pin. Bit 31 (ORE0) is the enable overflow flag for $\overline{\text{IRQ}}[0]$ ; bit 16 (ORE15) is the overflow flag for $\overline{\text{IRQ}}[15]$ .<br><br>0 Overflow request is disabled.<br>1 Overflow request is enabled. |
| 19<br>ORE12      | Overflow request enable 12. Enables the overflow request when an overflow occurs in the $\overline{\text{IRQ}}[12]$ pin. Bit 31 (ORE0) is the enable overflow flag for $\overline{\text{IRQ}}[0]$ ; bit 16 (ORE15) is the overflow flag for $\overline{\text{IRQ}}[15]$ .<br><br>0 Overflow request is disabled.<br>1 Overflow request is enabled. |
| 20<br>ORE11      | Overflow request enable 11. Enables the overflow request when an overflow occurs in the $\overline{\text{IRQ}}[11]$ pin. Bit 31 (ORE0) is the enable overflow flag for $\overline{\text{IRQ}}[0]$ ; bit 16 (ORE15) is the overflow flag for $\overline{\text{IRQ}}[15]$ .<br><br>0 Overflow request is disabled.<br>1 Overflow request is enabled. |
| 21<br>ORE10      | Overflow request enable 10. Enables the overflow request when an overflow occurs in the $\overline{\text{IRQ}}[10]$ pin. Bit 31 (ORE0) is the enable overflow flag for $\overline{\text{IRQ}}[0]$ ; bit 16 (ORE15) is the overflow flag for $\overline{\text{IRQ}}[15]$ .<br><br>0 Overflow request is disabled.<br>1 Overflow request is enabled. |
| 22<br>ORE9       | Overflow request enable 9. Enables the overflow request when an overflow occurs in the $\overline{\text{IRQ}}[9]$ pin. Bit 31 (ORE0) is the enable overflow flag for $\overline{\text{IRQ}}[0]$ ; bit 16 (ORE15) is the overflow flag for $\overline{\text{IRQ}}[15]$ .<br><br>0 Overflow request is disabled.<br>1 Overflow request is enabled.   |
| 23<br>ORE8       | Overflow request enable 8. Enables the overflow request when an overflow occurs in the $\overline{\text{IRQ}}[8]$ pin. Bit 31 (ORE0) is the enable overflow flag for $\overline{\text{IRQ}}[0]$ ; bit 16 (ORE15) is the overflow flag for $\overline{\text{IRQ}}[15]$ .<br><br>0 Overflow request is disabled.<br>1 Overflow request is enabled.   |
| 24<br>ORE7       | Overflow request enable 7. Enables the overflow request when an overflow occurs in the $\overline{\text{IRQ}}[7]$ pin. Bit 31 (ORE0) is the enable overflow flag for $\overline{\text{IRQ}}[0]$ ; bit 16 (ORE15) is the overflow flag for $\overline{\text{IRQ}}[15]$ .<br><br>0 Overflow request is disabled.<br>1 Overflow request is enabled.   |
| 25<br>ORE6       | Overflow request enable 6. Enables the overflow request when an overflow occurs in the $\overline{\text{IRQ}}[6]$ pin. Bit 31 (ORE0) is the enable overflow flag for $\overline{\text{IRQ}}[0]$ ; bit 16 (ORE15) is the overflow flag for $\overline{\text{IRQ}}[15]$ .                                                                            |

*Table continues on the next page...*

## SIU\_ORER field descriptions (continued)

| Field      | Description                                                                                                                                                                                                                                                                                                                               |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|            | 0 Overrun request is disabled.<br>1 Overrun request is enabled.                                                                                                                                                                                                                                                                           |
| 26<br>ORE5 | Overrun request enable 5. Enables the overrun request when an overrun occurs in the $\overline{\text{IRQ}}[5]$ pin. Bit 31 (ORE0) is the enable overrun flag for $\overline{\text{IRQ}}[0]$ ; bit 16 (ORE15) is the overrun flag for $\overline{\text{IRQ}}[15]$ .<br><br>0 Overrun request is disabled.<br>1 Overrun request is enabled. |
| 27<br>ORE4 | Overrun request enable 4. Enables the overrun request when an overrun occurs in the $\overline{\text{IRQ}}[4]$ pin. Bit 31 (ORE0) is the enable overrun flag for $\overline{\text{IRQ}}[0]$ ; bit 16 (ORE15) is the overrun flag for $\overline{\text{IRQ}}[15]$ .<br><br>0 Overrun request is disabled.<br>1 Overrun request is enabled. |
| 28<br>ORE3 | Overrun request enable 3. Enables the overrun request when an overrun occurs in the $\overline{\text{IRQ}}[3]$ pin. Bit 31 (ORE0) is the enable overrun flag for $\overline{\text{IRQ}}[0]$ ; bit 16 (ORE15) is the overrun flag for $\overline{\text{IRQ}}[15]$ .<br><br>0 Overrun request is disabled.<br>1 Overrun request is enabled. |
| 29<br>ORE2 | Overrun request enable 2. Enables the overrun request when an overrun occurs in the $\overline{\text{IRQ}}[2]$ pin. Bit 31 (ORE0) is the enable overrun flag for $\overline{\text{IRQ}}[0]$ ; bit 16 (ORE15) is the overrun flag for $\overline{\text{IRQ}}[15]$ .<br><br>0 Overrun request is disabled.<br>1 Overrun request is enabled. |
| 30<br>ORE1 | Overrun request enable 1. Enables the overrun request when an overrun occurs in the $\overline{\text{IRQ}}[1]$ pin. Bit 31 (ORE0) is the enable overrun flag for $\overline{\text{IRQ}}[0]$ ; bit 16 (ORE15) is the overrun flag for $\overline{\text{IRQ}}[15]$ .<br><br>0 Overrun request is disabled.<br>1 Overrun request is enabled. |
| 31<br>ORE0 | Overrun request enable 0. Enables the overrun request when an overrun occurs in the $\overline{\text{IRQ}}[0]$ pin. Bit 31 (ORE0) is the enable overrun flag for $\overline{\text{IRQ}}[0]$ ; bit 16 (ORE15) is the overrun flag for $\overline{\text{IRQ}}[15]$ .<br><br>0 Overrun request is disabled.<br>1 Overrun request is enabled. |

## 8.2.9 IRQ Rising-Edge Event Enable Register (SIU\_IREER)

The SIU\_IREER enables rising edge-triggered events on  $\overline{\text{IRQ}}[n]$ . Rising- and falling-edge events are enabled by setting the bits in SIU\_IREER and SIU\_IFEER.

SIU\_IREER bits used for NMI events are write once after a reset.

Address: 0h base + 28h offset = 28h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

## SIU\_IREE field descriptions

| Field            | Description                                                                                                                                                                                                                         |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>IREE_NMI8   | Core 0 NMI rising-edge event enable. Enables rising edge triggered events on the external NMI pin to Core 0. This bit can be written once after a reset.<br><br>0 Rising edge event is disabled.<br>1 Rising edge event is enabled. |
| 1<br>IREE_NMI7   | Core 1 NMI rising-edge event enable. Enables rising edge triggered events on the external NMI pin to Core 1. This bit can be written once after a reset.<br><br>0 Rising edge event is disabled.<br>1 Rising edge event is enabled. |
| 2–15<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                             |
| 16<br>IREE15     | IRQ rising-edge event enable 15. Enables rising edge triggered events on the external $\overline{\text{IRQ}}[15]$ pin.<br><br>0 Rising edge event is disabled.<br>1 Rising edge event is enabled.                                   |
| 17<br>IREE14     | IRQ rising-edge event enable 14. Enables rising edge triggered events on the external $\overline{\text{IRQ}}[14]$ pin.<br><br>0 Rising edge event is disabled.<br>1 Rising edge event is enabled.                                   |
| 18<br>IREE13     | IRQ rising-edge event enable 13. Enables rising edge triggered events on the external $\overline{\text{IRQ}}[13]$ pin.<br><br>0 Rising edge event is disabled.<br>1 Rising edge event is enabled.                                   |
| 19<br>IREE12     | IRQ rising-edge event enable 12. Enables rising edge triggered events on the external $\overline{\text{IRQ}}[12]$ pin.<br><br>0 Rising edge event is disabled.<br>1 Rising edge event is enabled.                                   |
| 20<br>IREE11     | IRQ rising-edge event enable 11. Enables rising edge triggered events on the external $\overline{\text{IRQ}}[11]$ pin.<br><br>0 Rising edge event is disabled.<br>1 Rising edge event is enabled.                                   |
| 21<br>IREE10     | IRQ rising-edge event enable 10. Enables rising edge triggered events on the external $\overline{\text{IRQ}}[10]$ pin.<br><br>0 Rising edge event is disabled.<br>1 Rising edge event is enabled.                                   |
| 22<br>IREE9      | IRQ rising-edge event enable 9. Enables rising edge triggered events on the external $\overline{\text{IRQ}}[9]$ pin.<br><br>0 Rising edge event is disabled.<br>1 Rising edge event is enabled.                                     |

Table continues on the next page...

**SIU\_IREER field descriptions (continued)**

| Field       | Description                                                                                                                                                                                 |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 23<br>IREE8 | IRQ rising-edge event enable 8. Enables rising edge triggered events on the external $\overline{\text{IRQ}}[8]$ pin.<br>0 Rising edge event is disabled.<br>1 Rising edge event is enabled. |
| 24<br>IREE7 | IRQ rising-edge event enable 7. Enables rising edge triggered events on the external $\overline{\text{IRQ}}[7]$ pin.<br>0 Rising edge event is disabled.<br>1 Rising edge event is enabled. |
| 25<br>IREE6 | IRQ rising-edge event enable 6. Enables rising edge triggered events on the external $\overline{\text{IRQ}}[6]$ pin.<br>0 Rising edge event is disabled.<br>1 Rising edge event is enabled. |
| 26<br>IREE5 | IRQ rising-edge event enable 5. Enables rising edge triggered events on the external $\overline{\text{IRQ}}[5]$ pin.<br>0 Rising edge event is disabled.<br>1 Rising edge event is enabled. |
| 27<br>IREE4 | IRQ rising-edge event enable 4. Enables rising edge triggered events on the external $\overline{\text{IRQ}}[4]$ pin.<br>0 Rising edge event is disabled.<br>1 Rising edge event is enabled. |
| 28<br>IREE3 | IRQ rising-edge event enable 3. Enables rising edge triggered events on the external $\overline{\text{IRQ}}[3]$ pin.<br>0 Rising edge event is disabled.<br>1 Rising edge event is enabled. |
| 29<br>IREE2 | IRQ rising-edge event enable 2. Enables rising edge triggered events on the external $\overline{\text{IRQ}}[2]$ pin.<br>0 Rising edge event is disabled.<br>1 Rising edge event is enabled. |
| 30<br>IREE1 | IRQ rising-edge event enable 1. Enables rising edge triggered events on the external $\overline{\text{IRQ}}[1]$ pin.<br>0 Rising edge event is disabled.<br>1 Rising edge event is enabled. |
| 31<br>IREE0 | IRQ rising-edge event enable 0. Enables rising edge triggered events on the external $\overline{\text{IRQ}}[0]$ pin.<br>0 Rising edge event is disabled.<br>1 Rising edge event is enabled. |

**8.2.10 IRQ Falling-Edge Event Enable Register (SIU\_IFEER)**

The SIU\_IFEER enables falling edge-triggered events on  $\overline{\text{IRQ}}[n]$ . Rising- and falling-edge events are enabled by setting the bits in both SIU\_IREER and SIU\_IFEER.

SIU\_IFEER bits used for NMI events are write once after a reset.



Address: 0h base + 2Ch offset = 2Ch

|       |           |           |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|-----------|-----------|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0         | 1         | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | IFEE_NMI8 | IFEE_NMI7 | 0 |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |           |           |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0         | 0         | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |         |         |         |         |         |         |        |        |        |        |        |        |        |        |        |        |
|-------|---------|---------|---------|---------|---------|---------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Bit   | 16      | 17      | 18      | 19      | 20      | 21      | 22     | 23     | 24     | 25     | 26     | 27     | 28     | 29     | 30     | 31     |
| R     | IFEEN15 | IFEEN14 | IFEEN13 | IFEEN12 | IFEEN11 | IFEEN10 | IFEEN9 | IFEEN8 | IFEEN7 | IFEEN6 | IFEEN5 | IFEEN4 | IFEEN3 | IFEEN2 | IFEEN1 | IFEEN0 |
| W     |         |         |         |         |         |         |        |        |        |        |        |        |        |        |        |        |
| Reset | 0       | 0       | 0       | 0       | 0       | 0       | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      |

**SIU\_IFEER field descriptions**

| Field            | Description                                                                                                                                                                                                                             |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>IFEE_NMI8   | Core 0 NMI falling-edge event enable. Enables falling edge triggered events on the external NMI pin to Core 0. This bit can be written once after a reset.<br><br>0 Falling edge event is disabled.<br>1 Falling edge event is enabled. |
| 1<br>IFEE_NMI7   | Core 1 NMI falling-edge event enable. Enables falling edge triggered events on the external NMI pin to Core 1. This bit can be written once after a reset.<br><br>0 Falling edge event is disabled.<br>1 Falling edge event is enabled. |
| 2–15<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                 |
| 16<br>IFEEN15    | IRQ falling-edge event enable 15. Enables falling edge triggered events on the external $\overline{\text{IRQ}}[15]$ pin.<br><br>0 Falling edge event is disabled.<br>1 Falling edge event is enabled.                                   |
| 17<br>IFEEN14    | IRQ falling-edge event enable 14. Enables falling edge triggered events on the external $\overline{\text{IRQ}}[14]$ pin.<br><br>0 Falling edge event is disabled.<br>1 Falling edge event is enabled.                                   |
| 18<br>IFEEN13    | IRQ falling-edge event enable 13. Enables falling edge triggered events on the external $\overline{\text{IRQ}}[13]$ pin.<br><br>0 Falling edge event is disabled.<br>1 Falling edge event is enabled.                                   |
| 19<br>IFEEN12    | IRQ falling-edge event enable 12. Enables falling edge triggered events on the external $\overline{\text{IRQ}}[12]$ pin.<br><br>0 Falling edge event is disabled.<br>1 Falling edge event is enabled.                                   |
| 20<br>IFEEN11    | IRQ falling-edge event enable 11. Enables falling edge triggered events on the external $\overline{\text{IRQ}}[11]$ pin.<br><br>0 Falling edge event is disabled.<br>1 Falling edge event is enabled.                                   |

*Table continues on the next page...*

## SIU\_IFEER field descriptions (continued)

| Field         | Description                                                                                                                                                                                       |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 21<br>IFEEN10 | IRQ falling-edge event enable 10. Enables falling edge triggered events on the external $\overline{\text{IRQ}}[10]$ pin.<br>0 Falling edge event is disabled.<br>1 Falling edge event is enabled. |
| 22<br>IFEEN9  | IRQ falling-edge event enable 9. Enables falling edge triggered events on the external $\overline{\text{IRQ}}[9]$ pin.<br>0 Falling edge event is disabled.<br>1 Falling edge event is enabled.   |
| 23<br>IFEEN8  | IRQ falling-edge event enable 8. Enables falling edge triggered events on the external $\overline{\text{IRQ}}[8]$ pin.<br>0 Falling edge event is disabled.<br>1 Falling edge event is enabled.   |
| 24<br>IFEEN7  | IRQ falling-edge event enable 7. Enables falling edge triggered events on the external $\overline{\text{IRQ}}[7]$ pin.<br>0 Falling edge event is disabled.<br>1 Falling edge event is enabled.   |
| 25<br>IFEEN6  | IRQ falling-edge event enable 6. Enables falling edge triggered events on the external $\overline{\text{IRQ}}[6]$ pin.<br>0 Falling edge event is disabled.<br>1 Falling edge event is enabled.   |
| 26<br>IFEEN5  | IRQ falling-edge event enable 5. Enables falling edge triggered events on the external $\overline{\text{IRQ}}[5]$ pin.<br>0 Falling edge event is disabled.<br>1 Falling edge event is enabled.   |
| 27<br>IFEEN4  | IRQ falling-edge event enable 4. Enables falling edge triggered events on the external $\overline{\text{IRQ}}[4]$ pin.<br>0 Falling edge event is disabled.<br>1 Falling edge event is enabled.   |
| 28<br>IFEEN3  | IRQ falling-edge event enable 3. Enables falling edge triggered events on the external $\overline{\text{IRQ}}[3]$ pin.<br>0 Falling edge event is disabled.<br>1 Falling edge event is enabled.   |
| 29<br>IFEEN2  | IRQ falling-edge event enable 2. Enables falling edge triggered events on the external $\overline{\text{IRQ}}[2]$ pin.<br>0 Falling edge event is disabled.<br>1 Falling edge event is enabled.   |
| 30<br>IFEEN1  | IRQ falling-edge event enable 1. Enables falling edge triggered events on the external $\overline{\text{IRQ}}[1]$ pin.<br>0 Falling edge event is disabled.<br>1 Falling edge event is enabled.   |
| 31<br>IFEEN0  | IRQ falling-edge event enable 0. Enables falling edge triggered events on the external $\overline{\text{IRQ}}[0]$ pin.<br>0 Falling edge event is disabled.<br>1 Falling edge event is enabled.   |

### 8.2.11 IRQ Digital Filter Register (SIU\_IDFR)

The SIU\_IDFR specifies the amount of digital filtering on  $\overline{\text{IRQ}}[0]-\overline{\text{IRQ}}[15]$  and NMI. The digital filter length field specifies the number of system clocks that define the period of the digital filter and the minimum time an IRQ or NMI signal must hold the active state to qualify as an edge-triggered event.

Address: 0h base + 30h offset = 30h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27  | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | DFL |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0  | 0  | 0  |    |

#### SIU\_IDFR field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–27<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                    |
| 28–31<br>DFL     | Digital filter length. Defines the digital filter period on the $\overline{\text{IRQ}}[n]$ and NMI inputs according to the following equation:<br><br>$\text{FilterPeriod} = (\text{PeripheralClockPeriod} \times 2^{\text{DFL}}) + 1(\text{PeripheralClockPeriod})$<br>For a 100 MHz peripheral clock, this gives a range of 20 ns to 328 ns. The minimum time of two system clocks accounts for synchronization of the $\overline{\text{IRQ}}$ and NMI input pins with the system clock. |

### 8.2.12 IRQ Filtered Input Register (SIU\_IFIR)

The SIU\_IFIR is a read only register where the filtered values of the NMI and  $\overline{\text{IRQ}}[0]-\overline{\text{IRQ}}[15]$  pins are captured.

Address: 0h base + 34h offset = 34h

|       |          |          |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|----------|----------|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0        | 1        | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | IFI_NMI8 | IFI_NMI7 | 0 |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |          |          |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0        | 0        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |       |       |       |       |       |       |      |      |      |      |      |      |      |      |      |      |
|-------|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| Bit   | 16    | 17    | 18    | 19    | 20    | 21    | 22   | 23   | 24   | 25   | 26   | 27   | 28   | 29   | 30   | 31   |
| R     | IFI15 | IFI14 | IFI13 | IFI12 | IFI11 | IFI10 | IFI9 | IFI8 | IFI7 | IFI6 | IFI5 | IFI4 | IFI3 | IFI2 | IFI1 | IFI0 |
| W     |       |       |       |       |       |       |      |      |      |      |      |      |      |      |      |      |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |

## SIU\_IFIR field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                   |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>IFI_NMI8    | NMI Filtered Input. This bit is set/cleared by the filtered NMI interrupt input.<br><br>0 A logic one has passed through the digital filter of the corresponding NMI pin.<br>1 A logic zero has passed through the digital filter of the corresponding NMI pin.                                                               |
| 1<br>IFI_NMI7    | NMI Filtered Input. This bit is set/cleared by the filtered NMI interrupt input.<br><br>0 A logic one has passed through the digital filter of the corresponding NMI pin.<br>1 A logic zero has passed through the digital filter of the corresponding NMI pin.                                                               |
| 2–15<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                       |
| 16<br>IFI15      | Filtered Input 15. This bit is set/cleared for the corresponding filtered $\overline{\text{IRQ}}$ pin.<br><br>0 A logic one has passed through the digital filter of the corresponding $\overline{\text{IRQ}}$ pin.<br>1 A logic zero has passed through the digital filter of the corresponding $\overline{\text{IRQ}}$ pin. |
| 17<br>IFI14      | Filtered Input 14. This bit is set/cleared for the corresponding filtered $\overline{\text{IRQ}}$ pin.<br><br>0 A logic one has passed through the digital filter of the corresponding $\overline{\text{IRQ}}$ pin.<br>1 A logic zero has passed through the digital filter of the corresponding $\overline{\text{IRQ}}$ pin. |
| 18<br>IFI13      | Filtered Input 13. This bit is set/cleared for the corresponding filtered $\overline{\text{IRQ}}$ pin.<br><br>0 A logic one has passed through the digital filter of the corresponding $\overline{\text{IRQ}}$ pin.<br>1 A logic zero has passed through the digital filter of the corresponding $\overline{\text{IRQ}}$ pin. |
| 19<br>IFI12      | Filtered Input 12. This bit is set/cleared for the corresponding filtered $\overline{\text{IRQ}}$ pin.<br><br>0 A logic one has passed through the digital filter of the corresponding $\overline{\text{IRQ}}$ pin.<br>1 A logic zero has passed through the digital filter of the corresponding $\overline{\text{IRQ}}$ pin. |
| 20<br>IFI11      | Filtered Input 11. This bit is set/cleared for the corresponding filtered $\overline{\text{IRQ}}$ pin.<br><br>0 A logic one has passed through the digital filter of the corresponding $\overline{\text{IRQ}}$ pin.<br>1 A logic zero has passed through the digital filter of the corresponding $\overline{\text{IRQ}}$ pin. |
| 21<br>IFI10      | Filtered Input 10. This bit is set/cleared for the corresponding filtered $\overline{\text{IRQ}}$ pin.<br><br>0 A logic one has passed through the digital filter of the corresponding $\overline{\text{IRQ}}$ pin.<br>1 A logic zero has passed through the digital filter of the corresponding $\overline{\text{IRQ}}$ pin. |
| 22<br>IFI9       | Filtered Input 9. This bit is set/cleared for the corresponding filtered $\overline{\text{IRQ}}$ pin.<br><br>0 A logic one has passed through the digital filter of the corresponding $\overline{\text{IRQ}}$ pin.<br>1 A logic zero has passed through the digital filter of the corresponding $\overline{\text{IRQ}}$ pin.  |
| 23<br>IFI8       | Filtered Input 8. This bit is set/cleared for the corresponding filtered $\overline{\text{IRQ}}$ pin.<br><br>0 A logic one has passed through the digital filter of the corresponding $\overline{\text{IRQ}}$ pin.<br>1 A logic zero has passed through the digital filter of the corresponding $\overline{\text{IRQ}}$ pin.  |
| 24<br>IFI7       | Filtered Input 7. This bit is set/cleared for the corresponding filtered $\overline{\text{IRQ}}$ pin.<br><br>0 A logic one has passed through the digital filter of the corresponding $\overline{\text{IRQ}}$ pin.<br>1 A logic zero has passed through the digital filter of the corresponding $\overline{\text{IRQ}}$ pin.  |
| 25<br>IFI6       | Filtered Input 6. This bit is set/cleared for the corresponding filtered $\overline{\text{IRQ}}$ pin.                                                                                                                                                                                                                         |

Table continues on the next page...

**SIU\_IFIR field descriptions (continued)**

| Field      | Description                                                                                                                                                                                                                                                                                                              |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|            | 0 A logic one has passed through the digital filter of the corresponding $\overline{\text{IRQ}}$ pin.<br>1 A logic zero has passed through the digital filter of the corresponding $\overline{\text{IRQ}}$ pin.                                                                                                          |
| 26<br>IFI5 | Filtered Input 5. This bit is set/cleared for the corresponding filtered $\overline{\text{IRQ}}$ pin.<br>0 A logic one has passed through the digital filter of the corresponding $\overline{\text{IRQ}}$ pin.<br>1 A logic zero has passed through the digital filter of the corresponding $\overline{\text{IRQ}}$ pin. |
| 27<br>IFI4 | Filtered Input 4. This bit is set/cleared for the corresponding filtered $\overline{\text{IRQ}}$ pin.<br>0 A logic one has passed through the digital filter of the corresponding $\overline{\text{IRQ}}$ pin.<br>1 A logic zero has passed through the digital filter of the corresponding $\overline{\text{IRQ}}$ pin. |
| 28<br>IFI3 | Filtered Input 3. This bit is set/cleared for the corresponding filtered $\overline{\text{IRQ}}$ pin.<br>0 A logic one has passed through the digital filter of the corresponding $\overline{\text{IRQ}}$ pin.<br>1 A logic zero has passed through the digital filter of the corresponding $\overline{\text{IRQ}}$ pin. |
| 29<br>IFI2 | Filtered Input 2. This bit is set/cleared for the corresponding filtered $\overline{\text{IRQ}}$ pin.<br>0 A logic one has passed through the digital filter of the corresponding $\overline{\text{IRQ}}$ pin.<br>1 A logic zero has passed through the digital filter of the corresponding $\overline{\text{IRQ}}$ pin. |
| 30<br>IFI1 | Filtered Input 1. This bit is set/cleared for the corresponding filtered $\overline{\text{IRQ}}$ pin.<br>0 A logic one has passed through the digital filter of the corresponding $\overline{\text{IRQ}}$ pin.<br>1 A logic zero has passed through the digital filter of the corresponding $\overline{\text{IRQ}}$ pin. |
| 31<br>IFI0 | Filtered Input 0. This bit is set/cleared for the corresponding filtered $\overline{\text{IRQ}}$ pin.<br>0 A logic one has passed through the digital filter of the corresponding $\overline{\text{IRQ}}$ pin.<br>1 A logic zero has passed through the digital filter of the corresponding $\overline{\text{IRQ}}$ pin. |

**8.2.13 Pad Configuration Register (SIU\_PCRn)**

These registers define the pad configuration for all configurable device pins by specifying the active function, direction, and electrical attributes for the pin. PCR information includes the register fields that are active for a given pin or group of pins and the reset state of the register.

The reset states of SIU\_PCRs reflect the values before the BAM program executes. The BAM program can change some pad configuration registers based on the reset configuration. Refer to the System Boot chapter for more information.

The SIU\_PCRs are 16-bit registers that are read from or written to as:

- 16-bit values aligned on 16-bit boundaries, or
- 32-bit values aligned on 32-bit address boundaries.

The PCR registers allow software to control the following range of electrical characteristics of the external pads:

- Weak pullup/down enable/disable
- Weak pullup/down selection
- Slew-rate selection for outputs
- Drive strength selection for outputs
- Input buffer enable (when direction is configured for output)
- Input hysteresis enable/disable
- Open drain/push-pull output selection
- Multiplexed function selection
- Data direction selection

NOTE

The actual characteristics available in any SIU\_PCR depend on the type of pad it controls.

All device pin names begin with the primary function, followed by the alternate function, and then GPIO. In some cases, the third function can be a secondary alternate, which supersedes the GPIO. Those exceptions are noted. For example, SIU\_PCR85 configures the CNTXB\_PCSC[3]\_GPIO[85] muxed signal, where CNTXB is the primary function and PCSC[3] is the alternate function. See [External signal descriptions, multiplexing, and attributes](#) for how to identify the source module for primary and alternate functions and the description of these signals. For an additional signal description, see the chapter for the module that uses this signal.

A number of input signals on the device have multiple external pins as input sources. Only one input source can be active at any time for these pins. A set of multiplexers controlled by the SIU\_IMUX0 to SIU\_IMUX12 registers selects which input drives the function.

For information about implemented fields and reset states of individual PCR<sub>s</sub>, see [Pad Configuration Register \(PCR\) details](#).

Address: 0h base + D6h offset + (2d × i), where i=0d to 416d

| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6   | 7   | 8   | 9  | 10  | 11  | 12  | 13 | 14  | 15  |
|-------|----|----|----|----|----|----|-----|-----|-----|----|-----|-----|-----|----|-----|-----|
| Read  | 0  |    |    | PA |    |    | OBE | IBE | DSC |    | ODE | HYS | SRC |    | WPE | WPS |
| Write |    |    |    |    |    |    |     |     |     |    |     |     |     |    |     |     |
| Reset | 0* | 0* | 0* | 0* | 0* | 0* | 0*  | 0*  | 0*  | 0* | 0*  | 0*  | 0*  | 0* | 0*  | 0*  |

\* Notes:  
•

SIU\_PCR<sub>n</sub> field descriptions

| Field           | Description                                                                             |
|-----------------|-----------------------------------------------------------------------------------------|
| 0–2<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |

Table continues on the next page...

SIU\_PCR<sub>n</sub> field descriptions (continued)

| Field        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3–5<br>PA    | <p>Pin assignment. Selects the function of a multiplexed pad. A separate port enable output signal from the SIU is asserted for each value of this register. The size of the field can be from 1 to 3 bits, depending on the amount of multiplexing on the pad.</p> <p><b>NOTE:</b> For any SIU_PCR that does not comply with these rules, the PA definition is given explicitly with the SIU_PCR definition.</p> <p>000 GPIO.<br/> 001 Primary function.<br/> 010 Alternate 1.<br/> 011 Alternate 2.<br/> 100 Alternate 3.<br/> 101 Invalid value.<br/> 110 Invalid value.<br/> 111 Invalid value.</p> |
| 6<br>OBE     | <p>Output buffer enable. Enables the pad as an output and drives the output buffer enable signal.</p> <p>0 Disable output buffer for the pad.<br/> 1 Enable output buffer for the pad.</p>                                                                                                                                                                                                                                                                                                                                                                                                              |
| 7<br>IBE     | <p>Input buffer enable. Enables the pad as an input and drives the input buffer enable signal.</p> <p>0 Disable input buffer for the pad.<br/> 1 Enable input buffer for the pad.</p>                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 8–9<br>DSC   | <p>Drive strength control. Controls the pad drive strength.</p> <p>Actual drive strength depends on the pad type and load. See the electrical specifications in the Data Sheet for this information.</p> <p>00 Minimum drive strength.<br/> 01 Medium drive strength.<br/> 10 Medium-high drive strength.<br/> 11 Maximum drive strength.</p>                                                                                                                                                                                                                                                           |
| 10<br>ODE    | <p>Open drain output enable. Controls output driver configuration for the pads. Either open drain or push/pull driver configurations can be selected. This feature applies to output pins only.</p> <p>0 Disable open drain for the pad (push/pull driver enabled).<br/> 1 Enable open drain for the pad.</p>                                                                                                                                                                                                                                                                                           |
| 11<br>HYS    | <p>Input hysteresis. Controls whether hysteresis is enabled for the pad.</p> <p>0 Disable hysteresis for the pad.<br/> 1 Enable hysteresis for the pad.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 12–13<br>SRC | <p>Slew rate control. Controls slew rate for the pad. Slew rate control pertains to pins with slow or medium I/O pad types, and the output signals are driven according to the value of this field. Actual slew rate depends on the pad type and load. See the electrical specifications in the Data Sheet for this information.</p> <p>00 Half drive strength with slew control enabled.<br/> 01 Full drive strength with slew control enabled.<br/> 10 Half drive strength with slew control disabled.<br/> 11 Full drive strength with slew control disabled.</p>                                    |
| 14<br>WPE    | <p>Weak pull up/down enable. Controls whether the weak pullup/down devices are enabled/disabled for the pad. Pullup/down devices are enabled by default.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                            |

*Table continues on the next page...*

**SIU\_PCR<sub>n</sub> field descriptions (continued)**

| Field     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|           | 0 Disable weak pull device for the pad.<br>1 Enable weak pull device for the pad.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 15<br>WPS | Weak pull up/down select. Controls whether weak pullup or weak pulldown devices are used for the pad when weak pullup/down devices are enabled. During reset, the weak pull up/down behavior is controlled by the WKPCFG bit of the Miscellaneous DCF client, or by the WKPCFG pin when operating in legacy mode ((LEG=1 in the Miscellaneous DCF client). The WPS bit determines whether weak pullup or pulldown devices are used after reset, or for pads in which the WKPCFG pin does not determine the reset weak pullup/down state.<br><br>0 Pulldown is enabled for the pad.<br>1 Pullup is enabled for the pad. |

**8.2.14 GPIO Pin Data Output Register (SIU\_GPDO<sub>n</sub>)**

Each SIU\_GPDO<sub>n</sub> register specifies the output data for the function assigned to the GPIO[*n*] pin. The *n* notation in the SIU\_GPDO<sub>n</sub> register name relate to the [*n*] in GPIO[*n*] signal name. For example, SIU\_GPDO246 contains the PDO246 bit for CNTXD\_GPIO246. The address for a GPDO pin is the GPIO number plus an offset of SIU\_BASE + 0600h. Gaps exist in the memory map where registers correspond to GPIO pins that are not implemented in the package.

Software writes to the SIU\_GPDO<sub>n</sub> registers to drive data out on the external pin. Each register drives one external pin, which allows independent control of the pin. Writes to the SIU\_GPDO<sub>n</sub> register will not be reflected to the pin when a primary or alternate function is assigned, or when an input function is assigned to the pin.

If the direction of a GPIO signal changes from input to output, the SIU\_GPDO<sub>n</sub> register value is automatically driven out to the external pin without the need for a software update.

Writes to an SIU\_GPDO<sub>n</sub> register have no effect when a primary or alternate function is assigned.

Address: 0h base + 64Bh offset + (1d × *i*), where *i*=0d to 416d

|       |   |   |   |   |   |   |                  |   |
|-------|---|---|---|---|---|---|------------------|---|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6                | 7 |
| Read  | 0 |   |   |   |   |   | PDO <sub>n</sub> |   |
| Write |   |   |   |   |   |   |                  |   |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0                | 0 |

**SIU\_GPDO<sub>n</sub> field descriptions**

| Field           | Description                                                                             |
|-----------------|-----------------------------------------------------------------------------------------|
| 0–6<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |

*Table continues on the next page...*



**SIU\_GPDO<sub>n</sub> field descriptions (continued)**

| Field                 | Description                                                                                                                                                                                                                                                                                                                                 |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7<br>PDO <sub>n</sub> | Pin Data Out. This bit stores the data to be driven out on the external GPIO. If the register is read, it returns the value written.<br><br>0 Logic low value is driven to the external GPIO pin when the pin is configured as an output.<br>1 Logic high value is driven to the external GPIO pin when the pin is configured as an output. |

**8.2.15 GPIO Pin Data Input Register (legacy) (SIU\_GPDI<sub>n</sub>L)****NOTE**

These registers are implemented for legacy purposes and are limited to 256 general purpose input registers. The GPDI address for a particular pin is the GPIO number plus an offset of SIU\_BASE + 0800h. For full access to all 512 general purpose input registers, the SIU\_GPDI<sub>n</sub> registers starting at address offset 0E00h should be used.

Each SIU\_GPDI<sub>n</sub> register specifies the input state for the GPDI[*n*] pin. The *n* notation in the SIU\_GPDI<sub>n</sub> register name relates to the [*n*] in GPIO[*n*] signal name. For example, SIU\_GPDI246 contains the PDI246 bit for CNTXD\_GPIO246. Gaps exist in the memory map where registers correspond to GPIO pins that are not implemented in the package.

Software reads the SIU\_GPDI<sub>n</sub> registers to get the input state of the external GPIO pin. Each GPDI register contains the input state of one external GPIO pin. If a GPDI register is configured as output, and the input buffer enable bit is set to one in the PCR register, the SIU\_GPDI<sub>n</sub> register reflects the state of the output pin.

Address: 0h base + 84Bh offset + (1d × i), where i=0d to 178d

|       |   |   |   |   |   |   |                  |   |
|-------|---|---|---|---|---|---|------------------|---|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6                | 7 |
| Read  | 0 |   |   |   |   |   | PDI <sub>n</sub> |   |
| Write |   |   |   |   |   |   |                  |   |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0                | 0 |

**SIU\_GPDI<sub>n</sub>L field descriptions**

| Field                 | Description                                                                                                                                                                        |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–6<br>Reserved       | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                            |
| 7<br>PDI <sub>n</sub> | Pin Data In. This bit reflects the state on the external GPIO/GPI pin. If PCR <sub>n</sub> [IBE] = 1, then:<br><br>0 Signal on pin is logic zero.<br>1 Signal on pin is logic one. |

## 8.2.16 External IRQ Input Select Register (SIU\_EIISR)

The SIU\_EIISR selects the source for the external interrupt/DMA inputs.

Address: 0h base + 904h offset = 904h

|       |        |   |        |   |        |   |        |   |        |   |        |    |       |    |       |    |
|-------|--------|---|--------|---|--------|---|--------|---|--------|---|--------|----|-------|----|-------|----|
| Bit   | 0      | 1 | 2      | 3 | 4      | 5 | 6      | 7 | 8      | 9 | 10     | 11 | 12    | 13 | 14    | 15 |
| R     |        |   |        |   |        |   |        |   |        |   |        |    |       |    |       |    |
| W     |        |   |        |   |        |   |        |   |        |   |        |    |       |    |       |    |
|       | ESEL15 |   | ESEL14 |   | ESEL13 |   | ESEL12 |   | ESEL11 |   | ESEL10 |    | ESEL9 |    | ESEL8 |    |
| Reset | 0      | 0 | 0      | 0 | 0      | 0 | 0      | 0 | 0      | 0 | 0      | 0  | 0     | 0  | 0     | 0  |

|       |       |    |       |    |       |    |       |    |       |    |       |    |       |    |       |    |
|-------|-------|----|-------|----|-------|----|-------|----|-------|----|-------|----|-------|----|-------|----|
| Bit   | 16    | 17 | 18    | 19 | 20    | 21 | 22    | 23 | 24    | 25 | 26    | 27 | 28    | 29 | 30    | 31 |
| R     |       |    |       |    |       |    |       |    |       |    |       |    |       |    |       |    |
| W     |       |    |       |    |       |    |       |    |       |    |       |    |       |    |       |    |
|       | ESEL7 |    | ESEL6 |    | ESEL5 |    | ESEL4 |    | ESEL3 |    | ESEL2 |    | ESEL1 |    | ESEL0 |    |
| Reset | 0     | 0  | 0     | 0  | 0     | 0  | 0     | 0  | 0     | 0  | 0     | 0  | 0     | 0  | 0     | 0  |

### SIU\_EIISR field descriptions

| Field         | Description                                                                                                                                                                                                                                      |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–1<br>ESEL15 | External IRQ input select 15. Specifies the source for $\overline{\text{IRQ}}[15]$ .<br>00 $\overline{\text{IRQ}}[15]$ input pin.<br>01 DSPI_B[15] deserialized input.<br>10 DSPI_C[0] deserialized input.<br>11 DSPI_D[1] deserialized input.   |
| 2–3<br>ESEL14 | External IRQ input select 14. Specifies the source for $\overline{\text{IRQ}}[14]$ .<br>00 $\overline{\text{IRQ}}[14]$ input pin.<br>01 DSPI_B[14] deserialized input.<br>10 DSPI_C[15] deserialized input.<br>11 DSPI_D[0] deserialized input.  |
| 4–5<br>ESEL13 | External IRQ input select 13. Specifies the source for $\overline{\text{IRQ}}[13]$ .<br>00 $\overline{\text{IRQ}}[13]$ input pin.<br>01 DSPI_B[13] deserialized input.<br>10 DSPI_C[14] deserialized input.<br>11 DSPI_D[15] deserialized input. |
| 6–7<br>ESEL12 | External IRQ input select 12. Specifies the source for $\overline{\text{IRQ}}[12]$ .<br>00 $\overline{\text{IRQ}}[12]$ input pin.<br>01 DSPI_B[12] deserialized input.<br>10 DSPI_C[13] deserialized input.<br>11 DSPI_D[14] deserialized input. |
| 8–9<br>ESEL11 | External IRQ input select 11. Specifies the source for $\overline{\text{IRQ}}[11]$ .<br>00 $\overline{\text{IRQ}}[11]$ input pin.<br>01 DSPI_B[11] deserialized input.<br>10 DSPI_C[12] deserialized input.<br>11 DSPI_D[13] deserialized input. |

Table continues on the next page...

**SIU\_EIISR field descriptions (continued)**

| Field           | Description                                                                                                                                                                                                                                      |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 10–11<br>ESEL10 | External IRQ input select 10. Specifies the source for $\overline{\text{IRQ}}[10]$ .<br>00 $\overline{\text{IRQ}}[10]$ input pin.<br>01 DSPI_B[10] deserialized input.<br>10 DSPI_C[11] deserialized input.<br>11 DSPI_D[12] deserialized input. |
| 12–13<br>ESEL9  | External IRQ input select 9. Specifies the source for $\overline{\text{IRQ}}[9]$ .<br>00 $\overline{\text{IRQ}}[9]$ input pin.<br>01 DSPI_B[9] deserialized input.<br>10 DSPI_C[10] deserialized input.<br>11 DSPI_D[11] deserialized input.     |
| 14–15<br>ESEL8  | External IRQ input select 8. Specifies the source for $\overline{\text{IRQ}}[8]$ .<br>00 $\overline{\text{IRQ}}[8]$ input pin.<br>01 DSPI_B[8] deserialized input.<br>10 DSPI_C[9] deserialized input.<br>11 DSPI_D[10] deserialized input.      |
| 16–17<br>ESEL7  | External IRQ input select 7. Specifies the source for $\overline{\text{IRQ}}[7]$ .<br>00 $\overline{\text{IRQ}}[7]$ input pin.<br>01 DSPI_B[7] deserialized input.<br>10 DSPI_C[8] deserialized input.<br>11 DSPI_D[9] deserialized input.       |
| 18–19<br>ESEL6  | External IRQ input select 6. Specifies the source for $\overline{\text{IRQ}}[6]$ .<br>00 $\overline{\text{IRQ}}[6]$ input pin.<br>01 DSPI_B[6] deserialized input.<br>10 DSPI_C[7] deserialized input.<br>11 DSPI_D[8] deserialized input.       |
| 20–21<br>ESEL5  | External IRQ input select 5. Specifies the source for $\overline{\text{IRQ}}[5]$ .<br>00 $\overline{\text{IRQ}}[5]$ input pin.<br>01 DSPI_B[5] deserialized input.<br>10 DSPI_C[6] deserialized input.<br>11 DSPI_D[7] deserialized input.       |
| 22–23<br>ESEL4  | External IRQ input select 4. Specifies the source for $\overline{\text{IRQ}}[4]$ .<br>00 $\overline{\text{IRQ}}[4]$ input pin.<br>01 DSPI_B[4] deserialized input.<br>10 DSPI_C[5] deserialized input.<br>11 DSPI_D[6] deserialized input.       |
| 24–25<br>ESEL3  | External IRQ input select 3. Specifies the source for $\overline{\text{IRQ}}[3]$ .<br>00 $\overline{\text{IRQ}}[3]$ input pin.<br>01 DSPI_B[3] deserialized input.<br>10 DSPI_C[4] deserialized input.<br>11 DSPI_D[5] deserialized input.       |

*Table continues on the next page...*

**SIU\_EIISR field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 26–27<br>ESEL2 | External IRQ input select 2. Specifies the source for $\overline{\text{IRQ}}[2]$ .<br>00 $\overline{\text{IRQ}}[2]$ input pin.<br>01 DSPI_B[2] deserialized input.<br>10 DSPI_C[3] deserialized input.<br>11 DSPI_D[4] deserialized input. |
| 28–29<br>ESEL1 | External IRQ input select 1. Specifies the source for $\overline{\text{IRQ}}[1]$ .<br>00 $\overline{\text{IRQ}}[1]$ input pin.<br>01 DSPI_B[1] deserialized input.<br>10 DSPI_C[2] deserialized input.<br>11 eMIOS_1 channel 7 output.     |
| 30–31<br>ESEL0 | External IRQ input select 0. Specifies the source for $\overline{\text{IRQ}}[0]$ .<br>00 $\overline{\text{IRQ}}[0]$ input pin.<br>01 DSPI_B[0] deserialized input.<br>10 DSPI_C[1] deserialized input.<br>11 eMIOS_1 channel 6 output.     |

**8.2.17 DSPI Input Select Register (SIU\_DISR)**

The SIU\_DISR specifies the following operations for each DSPI:

- Data input source
- Slave select
- Clock input

Address: 0h base + 908h offset = 908h

|       |         |   |        |   |         |   |          |   |         |   |        |    |         |    |          |    |
|-------|---------|---|--------|---|---------|---|----------|---|---------|---|--------|----|---------|----|----------|----|
| Bit   | 0       | 1 | 2      | 3 | 4       | 5 | 6        | 7 | 8       | 9 | 10     | 11 | 12      | 13 | 14       | 15 |
| R     |         |   |        |   |         |   |          |   |         |   |        |    |         |    |          |    |
| W     |         |   |        |   |         |   |          |   |         |   |        |    |         |    |          |    |
|       | SINSELA |   | SSSELA |   | SCKSELA |   | Reserved |   | SINSELB |   | SSSELB |    | SCKSELB |    | Reserved |    |
| Reset | 0       | 0 | 0      | 0 | 0       | 0 | 0        | 0 | 0       | 0 | 0      | 0  | 0       | 0  | 0        | 0  |

|       |         |    |        |    |         |    |          |    |         |    |        |    |         |    |          |    |
|-------|---------|----|--------|----|---------|----|----------|----|---------|----|--------|----|---------|----|----------|----|
| Bit   | 16      | 17 | 18     | 19 | 20      | 21 | 22       | 23 | 24      | 25 | 26     | 27 | 28      | 29 | 30       | 31 |
| R     |         |    |        |    |         |    |          |    |         |    |        |    |         |    |          |    |
| W     |         |    |        |    |         |    |          |    |         |    |        |    |         |    |          |    |
|       | SINSELC |    | SSSELC |    | SCKSELC |    | Reserved |    | SINSELD |    | SSSELD |    | SCKSELD |    | Reserved |    |
| Reset | 0       | 0  | 0      | 0  | 0       | 0  | 0        | 0  | 0       | 0  | 0      | 0  | 0       | 0  | 0        | 0  |

**SIU\_DISR field descriptions**

| Field          | Description                                                                                                                                      |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–1<br>SINSELA | DSPI A data input select. Specifies the source of the DSPI A data input.<br>00 SINA_PCSC[2]_GPIO[94] pin.<br>01 SOUTB.<br>10 SOUTC.<br>11 SOUTD. |

Table continues on the next page...

**SIU\_DISR field descriptions (continued)**

| Field             | Description                                                                                                                                                                                           |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2–3<br>SSSELA     | DSPI A slave select input select. Specifies the source of the DSPI A slave select input.<br>00 PCSA[0]_PCSD[2]_GPIO[96] pin.<br>01 PCSB[0] (master).<br>10 PCSC[0] (master).<br>11 PCSD[0] (master).  |
| 4–5<br>SCKSELA    | DSPI A clock input select. Specifies the source of the DSPI A clock input.<br>00 SCKA_PCSC[1]_GPIO[93] pin.<br>01 SCKB (master).<br>10 SCKC (master).<br>11 SCKD (master).                            |
| 6–7<br>Reserved   | This field is reserved.<br>Always preserve this field's reset value.                                                                                                                                  |
| 8–9<br>SINSELB    | DSPI B data input select. Specifies the source of DSPI B data input.<br>00 SINB_GPIO[103] pin.<br>01 SOUTA.<br>10 SOUTC.<br>11 SOUTD.                                                                 |
| 10–11<br>SSSELB   | DSPI B slave select input select. Specifies the source of the DSPI B slave select input.<br>00 PCSB[0]_PCSD[2]_GPIO[105] pin.<br>01 PCSA[0] (master).<br>10 PCSC[0] (master).<br>11 PCSD[0] (master). |
| 12–13<br>SCKSELB  | DSPI B clock input select. Specifies the source of the DSPI B clock input.<br>00 SCKB_GPIO[102] pin.<br>01 SCKA (master).<br>10 SCKC (master).<br>11 SCKD (master).                                   |
| 14–15<br>Reserved | This field is reserved.<br>Always preserve this field's reset value.                                                                                                                                  |
| 16–17<br>SINSELC  | DSPI C data input select. Specifies the source of the DSPI C data input.<br>00 PCSB[3]_SINC_GPIO[108] pin.<br>01 SOUTA.<br>10 SOUTB.<br>11 SOUTD.                                                     |
| 18–19<br>SSSELC   | DSPI C slave select input select. Specifies the source of the DSPI C slave select input.<br>00 PCSB[5]_PCSC[0]_GPIO[110] pin.<br>01 PCSA[0] (master).<br>10 PCSB[0] (master).<br>11 PCSD[0] (master). |
| 20–21<br>SCKSELC  | DSPI C clock input select. Specifies the source of the DSPI C clock input when in slave mode.                                                                                                         |

*Table continues on the next page...*

**SIU\_DISR field descriptions (continued)**

| Field             | Description                                                                                                                                                                                     |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | 00 PCSB[4]_SCKC_GPIO[109] pin.<br>01 SCKA (master).<br>10 SCKB (master).<br>11 SCKD (master).                                                                                                   |
| 22–23<br>Reserved | This field is reserved.<br>Always preserve this field's reset value.                                                                                                                            |
| 24–25<br>SINSELD  | DSPI D data input select. Specifies the source of the DSPI D data input.<br>00 PCSA[3]_GPIO[99] pin.<br>01 SOUTA.<br>10 SOUTB.<br>11 SOUTC.                                                     |
| 26–27<br>SSSELD   | DSPI D slave select input select. Specifies the source of the DSPI D slave select input.<br>00 PCSB[1]_PCSD[0]_GPIO[106] pin.<br>01 PCSA0 (master).<br>10 PCSB0 (master).<br>11 PCSC0 (master). |
| 28–29<br>SCKSELD  | DSPI D clock input select. Specifies the source of the DSPI D clock input in slave mode.<br>00 PCSA[2]_SCKD_GPIO[98] pin.<br>01 Invalid value.<br>10 SCKB (master).<br>11 SCKC (master).        |
| 30–31<br>Reserved | This field is reserved.<br>Always preserve this field's reset value.                                                                                                                            |

**8.2.18 eQADC Command FIFO Trigger Source Registers (SIU\_ISEL4)**

The eQADC Command FIFO[5:2] Trigger Source Register (SIU\_ISEL4) is used to select a trigger source for a command FIFO. SIU\_ISEL4 is used to select a trigger source for command FIFO[5:2] in eQADC\_A. The cTSEL (combined Trigger Select) field is used to configure one of many possible trigger sources for each command FIFO.

To trigger the eQADC, the trigger source must change to the state that the input to the command FIFO has been programmed to recognize. A command FIFO trigger input can be programmed to recognize either rising or falling edges, and low or high gated trigger types.

**Table 8-1036. eQADC\_A Command FIFO 2 Trigger Sources**

| cTSEL_2_A |   |   |   |   |   |   | eQADC_A Trigger Inputs  |
|-----------|---|---|---|---|---|---|-------------------------|
| 0         | 0 | 0 | 0 | 0 | 0 | 0 | Not Connected (default) |
| 0         | 0 | 0 | 0 | 0 | 0 | 1 | RTI Trigger             |

*Table continues on the next page...*

**Table 8-1036. eQADC\_A Command FIFO 2 Trigger Sources (continued)**

| cTSEL_2_A |   |   |   |   |   |   | eQADC_A Trigger Inputs |
|-----------|---|---|---|---|---|---|------------------------|
| 0         | 0 | 0 | 0 | 0 | 1 | 0 | PIT0 Trigger           |
| 0         | 0 | 0 | 0 | 0 | 1 | 1 | PIT1 Trigger           |
| 0         | 0 | 0 | 0 | 1 | 0 | 0 | PIT2 Trigger           |
| 0         | 0 | 0 | 0 | 1 | 0 | 1 | PIT3 Trigger           |
| 0         | 0 | 0 | 0 | 1 | 1 | 0 | eTPUA14                |
| 0         | 0 | 0 | 0 | 1 | 1 | 1 | ETRIG1 pin             |
| 0         | 0 | 0 | 1 | 0 | x | x | Reserved               |
| 0         | 0 | 0 | 1 | 1 | 0 | 0 | eTPUA28                |
| 0         | 0 | 0 | 1 | 1 | 0 | 1 | eTPUA29                |
| 0         | 0 | 0 | 1 | 1 | 1 | 0 | eTPUA30                |
| 0         | 0 | 0 | 1 | 1 | 1 | 1 | eTPUA31                |
| 0         | 0 | 1 | 0 | 0 | 0 | 0 | eTPUB28                |
| 0         | 0 | 1 | 0 | 0 | 0 | 1 | eTPUB29                |
| 0         | 0 | 1 | 0 | 0 | 1 | 0 | eTPUB30                |
| 0         | 0 | 1 | 0 | 0 | 1 | 1 | eTPUB31                |
| 0         | 0 | 1 | 0 | 1 | 0 | 0 | eTPUC28                |
| 0         | 0 | 1 | 0 | 1 | 0 | 1 | eTPUC29                |
| 0         | 0 | 1 | 0 | 1 | 1 | 0 | eTPUC30                |
| 0         | 0 | 1 | 0 | 1 | 1 | 1 | eTPUC31                |
| 0         | 0 | 1 | 1 | 0 | 0 | 0 | eMIOS_0 channel 16     |
| 0         | 0 | 1 | 1 | 0 | 0 | 1 | eMIOS_0 channel 17     |
| 0         | 0 | 1 | 1 | 0 | 1 | 0 | eMIOS_0 channel 18     |
| 0         | 0 | 1 | 1 | 0 | 1 | 1 | eMIOS_0 channel 19     |
| 0         | 0 | 1 | 1 | 1 | 0 | 0 | eMIOS_0 channel 20     |
| 0         | 0 | 1 | 1 | 1 | 0 | 1 | eMIOS_0 channel 21     |
| 0         | 0 | 1 | 1 | 1 | 1 | 0 | eMIOS_0 channel 22     |
| 0         | 0 | 1 | 1 | 1 | 1 | 1 | eMIOS_0 channel 23     |
| 0         | 1 | x | x | x | x | x | eTPUA29                |
| 1         | 0 | x | x | x | x | x | eMIOS_1 channel 7      |
| 1         | 1 | x | x | x | x | x | ETRIG0 Pin             |

**Table 8-1037. eQADC\_A Command FIFO 3 Trigger Sources**

| cTSEL_3_A |   |   |   |   |   |   | eQADC_A Trigger Inputs  |
|-----------|---|---|---|---|---|---|-------------------------|
| 0         | 0 | 0 | 0 | 0 | 0 | 0 | Not Connected (default) |
| 0         | 0 | 0 | 0 | 0 | 0 | 1 | RTI Trigger             |
| 0         | 0 | 0 | 0 | 0 | 1 | 0 | PIT0 Trigger            |
| 0         | 0 | 0 | 0 | 0 | 1 | 1 | PIT1 Trigger            |
| 0         | 0 | 0 | 0 | 1 | 0 | 0 | PIT2 Trigger            |

Table continues on the next page...

**Table 8-1037. eQADC\_A Command FIFO 3 Trigger Sources (continued)**

| cTSEL_3_A |   |   |   |   |   |   | eQADC_A Trigger Inputs |
|-----------|---|---|---|---|---|---|------------------------|
| 0         | 0 | 0 | 0 | 1 | 0 | 1 | PIT3 Trigger           |
| 0         | 0 | 0 | 0 | 1 | 1 | 0 | eTPUA22                |
| 0         | 0 | 0 | 0 | 1 | 1 | 1 | ETRIG0 pin             |
| 0         | 0 | 0 | 1 | 0 | x | x | Reserved               |
| 0         | 0 | 0 | 1 | 1 | 0 | 0 | eTPUA28                |
| 0         | 0 | 0 | 1 | 1 | 0 | 1 | eTPUA29                |
| 0         | 0 | 0 | 1 | 1 | 1 | 0 | eTPUA30                |
| 0         | 0 | 0 | 1 | 1 | 1 | 1 | eTPUA31                |
| 0         | 0 | 1 | 0 | 0 | 0 | 0 | eTPUB28                |
| 0         | 0 | 1 | 0 | 0 | 0 | 1 | eTPUB29                |
| 0         | 0 | 1 | 0 | 0 | 1 | 0 | eTPUB30                |
| 0         | 0 | 1 | 0 | 0 | 1 | 1 | eTPUB31                |
| 0         | 0 | 1 | 0 | 1 | 0 | 0 | eTPUC28                |
| 0         | 0 | 1 | 0 | 1 | 0 | 1 | eTPUC29                |
| 0         | 0 | 1 | 0 | 1 | 1 | 0 | eTPUC30                |
| 0         | 0 | 1 | 0 | 1 | 1 | 1 | eTPUC31                |
| 0         | 0 | 1 | 1 | 0 | 0 | 0 | eMIOS_0 channel 16     |
| 0         | 0 | 1 | 1 | 0 | 0 | 1 | eMIOS_0 channel 17     |
| 0         | 0 | 1 | 1 | 0 | 1 | 0 | eMIOS_0 channel 18     |
| 0         | 0 | 1 | 1 | 0 | 1 | 1 | eMIOS_0 channel 19     |
| 0         | 0 | 1 | 1 | 1 | 0 | 0 | eMIOS_0 channel 20     |
| 0         | 0 | 1 | 1 | 1 | 0 | 1 | eMIOS_0 channel 21     |
| 0         | 0 | 1 | 1 | 1 | 1 | 0 | eMIOS_0 channel 22     |
| 0         | 0 | 1 | 1 | 1 | 1 | 1 | eMIOS_0 channel 23     |
| 0         | 1 | x | x | x | x | x | eTPUA28                |
| 1         | 0 | x | x | x | x | x | eMIOS_1 channel 6      |
| 1         | 1 | x | x | x | x | x | ETRIG1 Pin             |

**Table 8-1038. eQADC\_A Command FIFO 4 Trigger Sources**

| cTSEL_4_A |   |   |   |   |   |   | eQADC_A Trigger Inputs  |
|-----------|---|---|---|---|---|---|-------------------------|
| 0         | 0 | 0 | 0 | 0 | 0 | 0 | Not Connected (default) |
| 0         | 0 | 0 | 0 | 0 | 0 | 1 | RTI Trigger             |
| 0         | 0 | 0 | 0 | 0 | 1 | 0 | PIT0 Trigger            |
| 0         | 0 | 0 | 0 | 0 | 1 | 1 | PIT1 Trigger            |
| 0         | 0 | 0 | 0 | 1 | 0 | 0 | PIT2 Trigger            |
| 0         | 0 | 0 | 0 | 1 | 0 | 1 | PIT3 Trigger            |
| 0         | 0 | 0 | 0 | 1 | 1 | 0 | eTPUA30                 |
| 0         | 0 | 0 | 0 | 1 | 1 | 1 | ETRIG1 pin              |

Table continues on the next page...



**Table 8-1038. eQADC\_A Command FIFO 4 Trigger Sources (continued)**

| cTSEL_4_A |   |   |   |   |   |   | eQADC_A Trigger Inputs |
|-----------|---|---|---|---|---|---|------------------------|
| 0         | 0 | 0 | 1 | 0 | x | x | Reserved               |
| 0         | 0 | 0 | 1 | 1 | 0 | 0 | eTPUA28                |
| 0         | 0 | 0 | 1 | 1 | 0 | 1 | eTPUA29                |
| 0         | 0 | 0 | 1 | 1 | 1 | 0 | eTPUA30                |
| 0         | 0 | 0 | 1 | 1 | 1 | 1 | eTPUA31                |
| 0         | 0 | 1 | 0 | 0 | 0 | 0 | eTPUB28                |
| 0         | 0 | 1 | 0 | 0 | 0 | 1 | eTPUB29                |
| 0         | 0 | 1 | 0 | 0 | 1 | 0 | eTPUB30                |
| 0         | 0 | 1 | 0 | 0 | 1 | 1 | eTPUB31                |
| 0         | 0 | 1 | 0 | 1 | 0 | 0 | eTPUC28                |
| 0         | 0 | 1 | 0 | 1 | 0 | 1 | eTPUC29                |
| 0         | 0 | 1 | 0 | 1 | 1 | 0 | eTPUC30                |
| 0         | 0 | 1 | 0 | 1 | 1 | 1 | eTPUC31                |
| 0         | 0 | 1 | 1 | 0 | 0 | 0 | eMIOS_0 channel 16     |
| 0         | 0 | 1 | 1 | 0 | 0 | 1 | eMIOS_0 channel 17     |
| 0         | 0 | 1 | 1 | 0 | 1 | 0 | eMIOS_0 channel 18     |
| 0         | 0 | 1 | 1 | 0 | 1 | 1 | eMIOS_0 channel 19     |
| 0         | 0 | 1 | 1 | 1 | 0 | 0 | eMIOS_0 channel 20     |
| 0         | 0 | 1 | 1 | 1 | 0 | 1 | eMIOS_0 channel 21     |
| 0         | 0 | 1 | 1 | 1 | 1 | 0 | eMIOS_0 channel 22     |
| 0         | 0 | 1 | 1 | 1 | 1 | 1 | eMIOS_0 channel 23     |
| 0         | 1 | x | x | x | x | x | eTPUA27                |
| 1         | 0 | x | x | x | x | x | eMIOS_1 channel 5      |
| 1         | 1 | x | x | x | x | x | ETRIG0 Pin             |

**Table 8-1039. eQADC\_A Command FIFO 5 Trigger Sources**

| cTSEL_5_A |   |   |   |   |   |   | eQADC_A Trigger Inputs  |
|-----------|---|---|---|---|---|---|-------------------------|
| 0         | 0 | 0 | 0 | 0 | 0 | 0 | Not Connected (default) |
| 0         | 0 | 0 | 0 | 0 | 0 | 1 | RTI Trigger             |
| 0         | 0 | 0 | 0 | 0 | 1 | 0 | PIT0 Trigger            |
| 0         | 0 | 0 | 0 | 0 | 1 | 1 | PIT1 Trigger            |
| 0         | 0 | 0 | 0 | 1 | 0 | 0 | PIT2 Trigger            |
| 0         | 0 | 0 | 0 | 1 | 0 | 1 | PIT3 Trigger            |
| 0         | 0 | 0 | 0 | 1 | 1 | 0 | Reserved                |
| 0         | 0 | 0 | 0 | 1 | 1 | 1 | ETRIG0 pin              |
| 0         | 0 | 0 | 1 | 0 | x | x | Reserved                |
| 0         | 0 | 0 | 1 | 1 | 0 | 0 | eTPUA28                 |
| 0         | 0 | 0 | 1 | 1 | 0 | 1 | eTPUA29                 |

Table continues on the next page...

Table 8-1039. eQADC\_A Command FIFO 5 Trigger Sources (continued)

| cTSEL_5_A |   |   |   |   |   |   | eQADC_A Trigger Inputs |
|-----------|---|---|---|---|---|---|------------------------|
| 0         | 0 | 0 | 1 | 1 | 1 | 0 | eTPUA30                |
| 0         | 0 | 0 | 1 | 1 | 1 | 1 | eTPUA31                |
| 0         | 0 | 1 | 0 | 0 | 0 | 0 | eTPUB28                |
| 0         | 0 | 1 | 0 | 0 | 0 | 1 | eTPUB29                |
| 0         | 0 | 1 | 0 | 0 | 1 | 0 | eTPUB30                |
| 0         | 0 | 1 | 0 | 0 | 1 | 1 | eTPUB31                |
| 0         | 0 | 1 | 0 | 1 | 0 | 0 | eTPUC28                |
| 0         | 0 | 1 | 0 | 1 | 0 | 1 | eTPUC29                |
| 0         | 0 | 1 | 0 | 1 | 1 | 0 | eTPUC30                |
| 0         | 0 | 1 | 0 | 1 | 1 | 1 | eTPUC31                |
| 0         | 0 | 1 | 1 | 0 | 0 | 0 | eMIOS_0 channel 16     |
| 0         | 0 | 1 | 1 | 0 | 0 | 1 | eMIOS_0 channel 17     |
| 0         | 0 | 1 | 1 | 0 | 1 | 0 | eMIOS_0 channel 18     |
| 0         | 0 | 1 | 1 | 0 | 1 | 1 | eMIOS_0 channel 19     |
| 0         | 0 | 1 | 1 | 1 | 0 | 0 | eMIOS_0 channel 20     |
| 0         | 0 | 1 | 1 | 1 | 0 | 1 | eMIOS_0 channel 21     |
| 0         | 0 | 1 | 1 | 1 | 1 | 0 | eMIOS_0 channel 22     |
| 0         | 0 | 1 | 1 | 1 | 1 | 1 | eMIOS_0 channel 23     |
| 0         | 1 | x | x | x | x | x | eTPUA26                |
| 1         | 0 | x | x | x | x | x | eMIOS_1 channel 4      |
| 1         | 1 | x | x | x | x | x | ETRIG1 Pin             |

Address: 0h base + 910h offset = 910h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0 |   |   |   |   |   |   |   | 0 |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0  |    |    |    |    |    |    |    | 0  |    |    |    |    |    |    |    |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

## SIU\_ISEL4 field descriptions

| Field           | Description                                                                             |
|-----------------|-----------------------------------------------------------------------------------------|
| 0<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 1–7<br>CTSEL5_A | Trigger source selection for EQADC_A Command FIFO 5.                                    |
| 8<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0. |

Table continues on the next page...

**SIU\_ISEL4 field descriptions (continued)**

| Field             | Description                                                                             |
|-------------------|-----------------------------------------------------------------------------------------|
| 9–15<br>CTSEL4_A  | Trigger source selection for EQADC_A Command FIFO 4.                                    |
| 16<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 17–23<br>CTSEL3_A | Trigger source selection for EQADC_A Command FIFO 3.                                    |
| 24<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 25–31<br>CTSEL2_A | Trigger source selection for EQADC_A Command FIFO2.                                     |

**8.2.19 eQADC Command FIFO Trigger Source Registers (SIU\_ISEL5)**

The eQADC Command FIFO[1:0] Trigger Source Register (SIU\_ISEL5) is used to select a trigger source for a command FIFO. SIU\_ISEL5 is used to select a trigger source for command FIFO[1:0] in eQADC\_A. The cTSEL (combined Trigger Select) field is used to configure one of many possible trigger sources for each command FIFO.

To trigger the eQADC, the trigger source must change to the state that the input to the command FIFO has been programmed to recognize. A command FIFO trigger input can be programmed to recognize either rising or falling edges, and low or high gated trigger types.

**Table 8-1041. eQADC\_A Command FIFO 0 Trigger Sources**

| cTSEL_0_A |   |   |   |   |   |   | eQADC_A Trigger Inputs  |
|-----------|---|---|---|---|---|---|-------------------------|
| 0         | 0 | 0 | 0 | 0 | 0 | 0 | Not connected (default) |
| 0         | 0 | 0 | 0 | 0 | 0 | 1 | RTI Trigger             |
| 0         | 0 | 0 | 0 | 0 | 1 | 0 | PIT0 Trigger            |
| 0         | 0 | 0 | 0 | 0 | 1 | 1 | PIT1 Trigger            |
| 0         | 0 | 0 | 0 | 1 | 0 | 0 | PIT2 Trigger            |
| 0         | 0 | 0 | 0 | 1 | 0 | 1 | PIT3 Trigger            |
| 0         | 0 | 0 | 0 | 1 | 1 | 0 | Reserved                |
| 0         | 0 | 0 | 0 | 1 | 1 | 1 | ETRIG1 pin              |
| 0         | 0 | 0 | 1 | 0 | x | x | Reserved                |
| 0         | 0 | 0 | 1 | 1 | 0 | 0 | eTPUA28                 |
| 0         | 0 | 0 | 1 | 1 | 0 | 1 | eTPUA29                 |
| 0         | 0 | 0 | 1 | 1 | 1 | 0 | eTPUA30                 |
| 0         | 0 | 0 | 1 | 1 | 1 | 1 | eTPUA31                 |
| 0         | 0 | 1 | 0 | 0 | 0 | 0 | eTPUB28                 |
| 0         | 0 | 1 | 0 | 0 | 0 | 1 | eTPUB29                 |

Table continues on the next page...

**Table 8-1041. eQADC\_A Command FIFO 0 Trigger Sources (continued)**

| cTSEL_0_A |   |   |   |   |   |   | eQADC_A Trigger Inputs |
|-----------|---|---|---|---|---|---|------------------------|
| 0         | 0 | 1 | 0 | 0 | 1 | 0 | eTPUB30                |
| 0         | 0 | 1 | 0 | 0 | 1 | 1 | eTPUB31                |
| 0         | 0 | 1 | 0 | 1 | 0 | 0 | eTPUC28                |
| 0         | 0 | 1 | 0 | 1 | 0 | 1 | eTPUC29                |
| 0         | 0 | 1 | 0 | 1 | 1 | 0 | eTPUC30                |
| 0         | 0 | 1 | 0 | 1 | 1 | 1 | eTPUC31                |
| 0         | 0 | 1 | 1 | 0 | 0 | 0 | eMIOS_0 channel 16     |
| 0         | 0 | 1 | 1 | 0 | 0 | 1 | eMIOS_0 channel 17     |
| 0         | 0 | 1 | 1 | 0 | 1 | 0 | eMIOS_0 channel 18     |
| 0         | 0 | 1 | 1 | 0 | 1 | 1 | eMIOS_0 channel 19     |
| 0         | 0 | 1 | 1 | 1 | 0 | 0 | eMIOS_0 channel 20     |
| 0         | 0 | 1 | 1 | 1 | 0 | 1 | eMIOS_0 channel 21     |
| 0         | 0 | 1 | 1 | 1 | 1 | 0 | eMIOS_0 channel 22     |
| 0         | 0 | 1 | 1 | 1 | 1 | 1 | eMIOS_0 channel 23     |
| 0         | 1 | x | x | x | x | x | eTPUA30                |
| 1         | 0 | x | x | x | x | x | eMIOS_1 channel 2      |
| 1         | 1 | x | x | x | x | x | ETRIG0 Pin             |

**Table 8-1042. eQADC\_A Command FIFO 1 Trigger Sources**

| cTSEL_1_A |   |   |   |   |   |   | eQADC_A Trigger Inputs  |
|-----------|---|---|---|---|---|---|-------------------------|
| 0         | 0 | 0 | 0 | 0 | 0 | 0 | Not Connected (default) |
| 0         | 0 | 0 | 0 | 0 | 0 | 1 | RTI Trigger             |
| 0         | 0 | 0 | 0 | 0 | 1 | 0 | PIT0 Trigger            |
| 0         | 0 | 0 | 0 | 0 | 1 | 1 | PIT1 Trigger            |
| 0         | 0 | 0 | 0 | 1 | 0 | 0 | PIT2 Trigger            |
| 0         | 0 | 0 | 0 | 1 | 0 | 1 | PIT3 Trigger            |
| 0         | 0 | 0 | 0 | 1 | 1 | 0 | eTPUA7                  |
| 0         | 0 | 0 | 0 | 1 | 1 | 1 | ETRIG0 pin              |
| 0         | 0 | 0 | 1 | 0 | x | x | Reserved                |
| 0         | 0 | 0 | 1 | 1 | 0 | 0 | eTPUA28                 |
| 0         | 0 | 0 | 1 | 1 | 0 | 1 | eTPUA29                 |
| 0         | 0 | 0 | 1 | 1 | 1 | 0 | eTPUA30                 |
| 0         | 0 | 0 | 1 | 1 | 1 | 1 | eTPUA31                 |
| 0         | 0 | 1 | 0 | 0 | 0 | 0 | eTPUB28                 |
| 0         | 0 | 1 | 0 | 0 | 0 | 1 | eTPUB29                 |
| 0         | 0 | 1 | 0 | 0 | 1 | 0 | eTPUB30                 |
| 0         | 0 | 1 | 0 | 0 | 1 | 1 | eTPUB31                 |
| 0         | 0 | 1 | 0 | 1 | 0 | 0 | eTPUC28                 |

Table continues on the next page...

**Table 8-1042. eQADC\_A Command FIFO 1 Trigger Sources (continued)**

| cTSEL_1_A |   |   |   |   |   |   | eQADC_A Trigger Inputs |
|-----------|---|---|---|---|---|---|------------------------|
| 0         | 0 | 1 | 0 | 1 | 0 | 1 | eTPUC29                |
| 0         | 0 | 1 | 0 | 1 | 1 | 0 | eTPUC30                |
| 0         | 0 | 1 | 0 | 1 | 1 | 1 | eTPUC31                |
| 0         | 0 | 1 | 1 | 0 | 0 | 0 | eMIOS_0 channel 16     |
| 0         | 0 | 1 | 1 | 0 | 0 | 1 | eMIOS_0 channel 17     |
| 0         | 0 | 1 | 1 | 0 | 1 | 0 | eMIOS_0 channel 18     |
| 0         | 0 | 1 | 1 | 0 | 1 | 1 | eMIOS_0 channel 19     |
| 0         | 0 | 1 | 1 | 1 | 0 | 0 | eMIOS_0 channel 20     |
| 0         | 0 | 1 | 1 | 1 | 0 | 1 | eMIOS_0 channel 21     |
| 0         | 0 | 1 | 1 | 1 | 1 | 0 | eMIOS_0 channel 22     |
| 0         | 0 | 1 | 1 | 1 | 1 | 1 | eMIOS_0 channel 23     |
| 0         | 1 | x | x | x | x | x | eTPUA31                |
| 1         | 0 | x | x | x | x | x | eMIOS_1 channel 3      |
| 1         | 1 | x | x | x | x | x | ETRIG1 Pin             |

Address: 0h base + 914h offset = 914h

|       |    |          |    |    |    |    |    |    |    |          |    |    |    |    |    |    |
|-------|----|----------|----|----|----|----|----|----|----|----------|----|----|----|----|----|----|
| Bit   | 0  | 1        | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9        | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0  | CTSEL1_A |    |    |    |    |    |    | 0  | CTSEL0_A |    |    |    |    |    |    |
| W     |    |          |    |    |    |    |    |    |    |          |    |    |    |    |    |    |
| Reset | 0  | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0        | 0  | 0  | 0  | 0  | 0  | 0  |
| Bit   | 16 | 17       | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25       | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0  |          |    |    |    |    |    |    |    |          |    |    |    |    |    |    |
| W     |    |          |    |    |    |    |    |    |    |          |    |    |    |    |    |    |
| Reset | 0  | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0        | 0  | 0  | 0  | 0  | 0  | 0  |

**SIU\_ISEL5 field descriptions**

| Field             | Description                                                                             |
|-------------------|-----------------------------------------------------------------------------------------|
| 0<br>Reserved     | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 1–7<br>CTSEL1_A   | Trigger source selection for EQADC_A Command FIFO 1.                                    |
| 8<br>Reserved     | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 9–15<br>CTSEL0_A  | Trigger source selection for EQADC_A Command FIFO 0.                                    |
| 16–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |

## 8.2.20 eQADC Command FIFO Trigger Source Registers (SIU\_ISEL6)

The eQADC Command FIFO[5:2] Trigger Source Register (SIU\_ISEL6) is used to select a trigger source for a command FIFO. SIU\_ISEL6 is used to select a trigger source for command FIFO[5:2] in eQADC\_B. The cTSEL (combined Trigger Select) field is used to configure one of many possible trigger sources for each command FIFO.

To trigger the eQADC, the trigger source must change to the state that the input to the command FIFO has been programmed to recognize. A command FIFO trigger input can be programmed to recognize either rising or falling edges, and low or high gated trigger types.

**Table 8-1044. eQADC\_B Command FIFO 2 Trigger Sources**

| cTSEL_2_B |   |   |   |   |   |   | eQADC_B Trigger Inputs  |
|-----------|---|---|---|---|---|---|-------------------------|
| 0         | 0 | 0 | 0 | 0 | 0 | 0 | Not Connected (default) |
| 0         | 0 | 0 | 0 | 0 | 0 | 1 | RTI Trigger             |
| 0         | 0 | 0 | 0 | 0 | 1 | 0 | PIT0 Trigger            |
| 0         | 0 | 0 | 0 | 0 | 1 | 1 | PIT1 Trigger            |
| 0         | 0 | 0 | 0 | 1 | 0 | 0 | PIT2 Trigger            |
| 0         | 0 | 0 | 0 | 1 | 0 | 1 | PIT3 Trigger            |
| 0         | 0 | 0 | 0 | 1 | 1 | 0 | eTPUA14                 |
| 0         | 0 | 0 | 0 | 1 | 1 | 1 | ETRIG1 pin              |
| 0         | 0 | 0 | 1 | 0 | x | x | Reserved                |
| 0         | 0 | 0 | 1 | 1 | 0 | 0 | eTPUA28                 |
| 0         | 0 | 0 | 1 | 1 | 0 | 1 | eTPUA29                 |
| 0         | 0 | 0 | 1 | 1 | 1 | 0 | eTPUA30                 |
| 0         | 0 | 0 | 1 | 1 | 1 | 1 | eTPUA31                 |
| 0         | 0 | 1 | 0 | 0 | 0 | 0 | eTPUB28                 |
| 0         | 0 | 1 | 0 | 0 | 0 | 1 | eTPUB29                 |
| 0         | 0 | 1 | 0 | 0 | 1 | 0 | eTPUB30                 |
| 0         | 0 | 1 | 0 | 0 | 1 | 1 | eTPUB31                 |
| 0         | 0 | 1 | 0 | 1 | 0 | 0 | eTPUC28                 |
| 0         | 0 | 1 | 0 | 1 | 0 | 1 | eTPUC29                 |
| 0         | 0 | 1 | 0 | 1 | 1 | 0 | eTPUC30                 |
| 0         | 0 | 1 | 0 | 1 | 1 | 1 | eTPUC31                 |
| 0         | 0 | 1 | 1 | 0 | 0 | 0 | eMIOS_0 channel 16      |
| 0         | 0 | 1 | 1 | 0 | 0 | 1 | eMIOS_0 channel 17      |
| 0         | 0 | 1 | 1 | 0 | 1 | 0 | eMIOS_0 channel 18      |
| 0         | 0 | 1 | 1 | 0 | 1 | 1 | eMIOS_0 channel 19      |
| 0         | 0 | 1 | 1 | 1 | 0 | 0 | eMIOS_0 channel 20      |
| 0         | 0 | 1 | 1 | 1 | 0 | 1 | eMIOS_0 channel 21      |

Table continues on the next page...

**Table 8-1044. eQADC\_B Command FIFO 2 Trigger Sources (continued)**

| cTSEL_2_B |   |   |   |   |   |   | eQADC_B Trigger Inputs |
|-----------|---|---|---|---|---|---|------------------------|
| 0         | 0 | 1 | 1 | 1 | 1 | 0 | eMIOS_0 channel 22     |
| 0         | 0 | 1 | 1 | 1 | 1 | 1 | eMIOS_0 channel 23     |
| 0         | 1 | x | x | x | x | x | eTPUA29                |
| 1         | 0 | x | x | x | x | x | eMIOS_1 channel 7      |
| 1         | 1 | x | x | x | x | x | ETRIG0 Pin             |

**Table 8-1045. eQADC\_B Command FIFO 3 Trigger Sources**

| cTSEL_3_B |   |   |   |   |   |   | eQADC_B Trigger Inputs  |
|-----------|---|---|---|---|---|---|-------------------------|
| 0         | 0 | 0 | 0 | 0 | 0 | 0 | Not Connected (default) |
| 0         | 0 | 0 | 0 | 0 | 0 | 1 | RTI Trigger             |
| 0         | 0 | 0 | 0 | 0 | 1 | 0 | PIT0 Trigger            |
| 0         | 0 | 0 | 0 | 0 | 1 | 1 | PIT1 Trigger            |
| 0         | 0 | 0 | 0 | 1 | 0 | 0 | PIT2 Trigger            |
| 0         | 0 | 0 | 0 | 1 | 0 | 1 | PIT3 Trigger            |
| 0         | 0 | 0 | 0 | 1 | 1 | 0 | eTPUA22                 |
| 0         | 0 | 0 | 0 | 1 | 1 | 1 | ETRIG0 pin              |
| 0         | 0 | 0 | 1 | 0 | x | x | Reserved                |
| 0         | 0 | 0 | 1 | 1 | 0 | 0 | eTPUA28                 |
| 0         | 0 | 0 | 1 | 1 | 0 | 1 | eTPUA29                 |
| 0         | 0 | 0 | 1 | 1 | 1 | 0 | eTPUA30                 |
| 0         | 0 | 0 | 1 | 1 | 1 | 1 | eTPUA31                 |
| 0         | 0 | 1 | 0 | 0 | 0 | 0 | eTPUB28                 |
| 0         | 0 | 1 | 0 | 0 | 0 | 1 | eTPUB29                 |
| 0         | 0 | 1 | 0 | 0 | 1 | 0 | eTPUB30                 |
| 0         | 0 | 1 | 0 | 0 | 1 | 1 | eTPUB31                 |
| 0         | 0 | 1 | 0 | 1 | 0 | 0 | eTPUC28                 |
| 0         | 0 | 1 | 0 | 1 | 0 | 1 | eTPUC29                 |
| 0         | 0 | 1 | 0 | 1 | 1 | 0 | eTPUC30                 |
| 0         | 0 | 1 | 0 | 1 | 1 | 1 | eTPUC31                 |
| 0         | 0 | 1 | 1 | 0 | 0 | 0 | eMIOS_0 channel 16      |
| 0         | 0 | 1 | 1 | 0 | 0 | 1 | eMIOS_0 channel 17      |
| 0         | 0 | 1 | 1 | 0 | 1 | 0 | eMIOS_0 channel 18      |
| 0         | 0 | 1 | 1 | 0 | 1 | 1 | eMIOS_0 channel 19      |
| 0         | 0 | 1 | 1 | 1 | 0 | 0 | eMIOS_0 channel 20      |
| 0         | 0 | 1 | 1 | 1 | 0 | 1 | eMIOS_0 channel 21      |
| 0         | 0 | 1 | 1 | 1 | 1 | 0 | eMIOS_0 channel 22      |
| 0         | 0 | 1 | 1 | 1 | 1 | 1 | eMIOS_0 channel 23      |
| 0         | 1 | x | x | x | x | x | eTPUA28                 |

Table continues on the next page...

**Table 8-1045. eQADC\_B Command FIFO 3 Trigger Sources (continued)**

| cTSEL_3_B |   |   |   |   |   |   | eQADC_B Trigger Inputs |
|-----------|---|---|---|---|---|---|------------------------|
| 1         | 0 | x | x | x | x | x | eMIOS_1 channel 6      |
| 1         | 1 | x | x | x | x | x | ETRIG1 Pin             |

**Table 8-1046. eQADC\_B Command FIFO 4 Trigger Sources**

| cTSEL_4_B |   |   |   |   |   |   | eQADC_B Trigger Inputs  |
|-----------|---|---|---|---|---|---|-------------------------|
| 0         | 0 | 0 | 0 | 0 | 0 | 0 | Not Connected (default) |
| 0         | 0 | 0 | 0 | 0 | 0 | 1 | RTI Trigger             |
| 0         | 0 | 0 | 0 | 0 | 1 | 0 | PIT0 Trigger            |
| 0         | 0 | 0 | 0 | 0 | 1 | 1 | PIT1 Trigger            |
| 0         | 0 | 0 | 0 | 1 | 0 | 0 | PIT2 Trigger            |
| 0         | 0 | 0 | 0 | 1 | 0 | 1 | PIT3 Trigger            |
| 0         | 0 | 0 | 0 | 1 | 1 | 0 | eTPUA30                 |
| 0         | 0 | 0 | 0 | 1 | 1 | 1 | ETRIG1 pin              |
| 0         | 0 | 0 | 1 | 0 | x | x | Reserved                |
| 0         | 0 | 0 | 1 | 1 | 0 | 0 | eTPUA28                 |
| 0         | 0 | 0 | 1 | 1 | 0 | 1 | eTPUA29                 |
| 0         | 0 | 0 | 1 | 1 | 1 | 0 | eTPUA30                 |
| 0         | 0 | 0 | 1 | 1 | 1 | 1 | eTPUA31                 |
| 0         | 0 | 1 | 0 | 0 | 0 | 0 | eTPUB28                 |
| 0         | 0 | 1 | 0 | 0 | 0 | 1 | eTPUB29                 |
| 0         | 0 | 1 | 0 | 0 | 1 | 0 | eTPUB30                 |
| 0         | 0 | 1 | 0 | 0 | 1 | 1 | eTPUB31                 |
| 0         | 0 | 1 | 0 | 1 | 0 | 0 | eTPUC28                 |
| 0         | 0 | 1 | 0 | 1 | 0 | 1 | eTPUC29                 |
| 0         | 0 | 1 | 0 | 1 | 1 | 0 | eTPUC30                 |
| 0         | 0 | 1 | 0 | 1 | 1 | 1 | eTPUC31                 |
| 0         | 0 | 1 | 1 | 0 | 0 | 0 | eMIOS_0 channel 16      |
| 0         | 0 | 1 | 1 | 0 | 0 | 1 | eMIOS_0 channel 17      |
| 0         | 0 | 1 | 1 | 0 | 1 | 0 | eMIOS_0 channel 18      |
| 0         | 0 | 1 | 1 | 0 | 1 | 1 | eMIOS_0 channel 19      |
| 0         | 0 | 1 | 1 | 1 | 0 | 0 | eMIOS_0 channel 20      |
| 0         | 0 | 1 | 1 | 1 | 0 | 1 | eMIOS_0 channel 21      |
| 0         | 0 | 1 | 1 | 1 | 1 | 0 | eMIOS_0 channel 22      |
| 0         | 0 | 1 | 1 | 1 | 1 | 1 | eMIOS_0 channel 23      |
| 0         | 1 | x | x | x | x | x | eTPUA27                 |
| 1         | 0 | x | x | x | x | x | eMIOS_1 channel 5       |
| 1         | 1 | x | x | x | x | x | ETRIG0 Pin              |



Table 8-1047. eQADC\_B Command FIFO 5 Trigger Sources

| cTSEL_5_B |   |   |   |   |   |   | eQADC_B Trigger Inputs  |
|-----------|---|---|---|---|---|---|-------------------------|
| 0         | 0 | 0 | 0 | 0 | 0 | 0 | Not Connected (default) |
| 0         | 0 | 0 | 0 | 0 | 0 | 1 | RTI Trigger             |
| 0         | 0 | 0 | 0 | 0 | 1 | 0 | PIT0 Trigger            |
| 0         | 0 | 0 | 0 | 0 | 1 | 1 | PIT1 Trigger            |
| 0         | 0 | 0 | 0 | 1 | 0 | 0 | PIT2 Trigger            |
| 0         | 0 | 0 | 0 | 1 | 0 | 1 | PIT3 Trigger            |
| 0         | 0 | 0 | 0 | 1 | 1 | 0 | Reserved                |
| 0         | 0 | 0 | 0 | 1 | 1 | 1 | ETRIG0 pin              |
| 0         | 0 | 0 | 1 | 0 | x | x | Reserved                |
| 0         | 0 | 0 | 1 | 1 | 0 | 0 | eTPUA28                 |
| 0         | 0 | 0 | 1 | 1 | 0 | 1 | eTPUA29                 |
| 0         | 0 | 0 | 1 | 1 | 1 | 0 | eTPUA30                 |
| 0         | 0 | 0 | 1 | 1 | 1 | 1 | eTPUA31                 |
| 0         | 0 | 1 | 0 | 0 | 0 | 0 | eTPUB28                 |
| 0         | 0 | 1 | 0 | 0 | 0 | 1 | eTPUB29                 |
| 0         | 0 | 1 | 0 | 0 | 1 | 0 | eTPUB30                 |
| 0         | 0 | 1 | 0 | 0 | 1 | 1 | eTPUB31                 |
| 0         | 0 | 1 | 0 | 1 | 0 | 0 | eTPUC28                 |
| 0         | 0 | 1 | 0 | 1 | 0 | 1 | eTPUC29                 |
| 0         | 0 | 1 | 0 | 1 | 1 | 0 | eTPUC30                 |
| 0         | 0 | 1 | 0 | 1 | 1 | 1 | eTPUC31                 |
| 0         | 0 | 1 | 1 | 0 | 0 | 0 | eMIOS_0 channel 16      |
| 0         | 0 | 1 | 1 | 0 | 0 | 1 | eMIOS_0 channel 17      |
| 0         | 0 | 1 | 1 | 0 | 1 | 0 | eMIOS_0 channel 18      |
| 0         | 0 | 1 | 1 | 0 | 1 | 1 | eMIOS_0 channel 19      |
| 0         | 0 | 1 | 1 | 1 | 0 | 0 | eMIOS_0 channel 20      |
| 0         | 0 | 1 | 1 | 1 | 0 | 1 | eMIOS_0 channel 21      |
| 0         | 0 | 1 | 1 | 1 | 1 | 0 | eMIOS_0 channel 22      |
| 0         | 0 | 1 | 1 | 1 | 1 | 1 | eMIOS_0 channel 23      |
| 0         | 1 | x | x | x | x | x | eTPUA26                 |
| 1         | 0 | x | x | x | x | x | eMIOS_1 channel 4       |
| 1         | 1 | x | x | x | x | x | ETRIG1 Pin              |

Address: 0h base + 918h offset = 918h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0 |   |   |   |   |   |   |   | 0 |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |          |    |    |    |    |    |    |    |          |    |    |    |    |    |    |
|-------|----|----------|----|----|----|----|----|----|----|----------|----|----|----|----|----|----|
| Bit   | 16 | 17       | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25       | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0  | CTSEL3_B |    |    |    |    |    |    | 0  | CTSEL2_B |    |    |    |    |    |    |
| W     |    |          |    |    |    |    |    |    |    |          |    |    |    |    |    |    |
| Reset | 0  | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0        | 0  | 0  | 0  | 0  | 0  | 0  |

**SIU\_ISEL6 field descriptions**

| Field             | Description                                                                             |
|-------------------|-----------------------------------------------------------------------------------------|
| 0<br>Reserved     | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 1–7<br>CTSEL5_B   | Trigger source selection for EQADC_B Command FIFO 5.                                    |
| 8<br>Reserved     | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 9–15<br>CTSEL4_B  | Trigger source selection for EQADC_B Command FIFO 4.                                    |
| 16<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 17–23<br>CTSEL3_B | Trigger source selection for EQADC_B Command FIFO 3.                                    |
| 24<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 25–31<br>CTSEL2_B | Trigger source selection for EQADC_B Command FIFO 2.                                    |

**8.2.21 eQADC Command FIFO Trigger Source Registers (SIU\_ISEL7)**

The eQADC Command FIFO[1:0] Trigger Source Register (SIU\_ISEL7) is used to select a trigger source for a command FIFO. SIU\_ISEL7 is used to select a trigger source for command FIFO[1:0] in eQADC\_B. The cTSEL (combined Trigger Select) field is used to configure one of many possible trigger sources for each command FIFO.

To trigger the eQADC, the trigger source must change to the state that the input to the command FIFO has been programmed to recognize. A command FIFO trigger input can be programmed to recognize either rising or falling edges, and low or high gated trigger types.

**Table 8-1049. eQADC\_B Command FIFO 0 Trigger Sources**

| cTSEL_0_B |   |   |   |   |   |   | eQADC_B Trigger Inputs  |
|-----------|---|---|---|---|---|---|-------------------------|
| 0         | 0 | 0 | 0 | 0 | 0 | 0 | Not Connected (default) |
| 0         | 0 | 0 | 0 | 0 | 0 | 1 | RTI Trigger             |
| 0         | 0 | 0 | 0 | 0 | 1 | 0 | PIT0 Trigger            |
| 0         | 0 | 0 | 0 | 0 | 1 | 1 | PIT1 Trigger            |
| 0         | 0 | 0 | 0 | 1 | 0 | 0 | PIT2 Trigger            |

*Table continues on the next page...*

**Table 8-1049. eQADC\_B Command FIFO 0 Trigger Sources (continued)**

| cTSEL_0_B |   |   |   |   |   |   | eQADC_B Trigger Inputs |
|-----------|---|---|---|---|---|---|------------------------|
| 0         | 0 | 0 | 0 | 1 | 0 | 1 | PIT3 Trigger           |
| 0         | 0 | 0 | 0 | 1 | 1 | 0 | Reserved               |
| 0         | 0 | 0 | 0 | 1 | 1 | 1 | ETRIG1 pin             |
| 0         | 0 | 0 | 1 | 0 | x | x | Reserved               |
| 0         | 0 | 0 | 1 | 1 | 0 | 0 | eTPUA28                |
| 0         | 0 | 0 | 1 | 1 | 0 | 1 | eTPUA29                |
| 0         | 0 | 0 | 1 | 1 | 1 | 0 | eTPUA30                |
| 0         | 0 | 0 | 1 | 1 | 1 | 1 | eTPUA31                |
| 0         | 0 | 1 | 0 | 0 | 0 | 0 | eTPUB28                |
| 0         | 0 | 1 | 0 | 0 | 0 | 1 | eTPUB29                |
| 0         | 0 | 1 | 0 | 0 | 1 | 0 | eTPUB30                |
| 0         | 0 | 1 | 0 | 0 | 1 | 1 | eTPUB31                |
| 0         | 0 | 1 | 0 | 1 | 0 | 0 | eTPUC28                |
| 0         | 0 | 1 | 0 | 1 | 0 | 1 | eTPUC29                |
| 0         | 0 | 1 | 0 | 1 | 1 | 0 | eTPUC30                |
| 0         | 0 | 1 | 0 | 1 | 1 | 1 | eTPUC31                |
| 0         | 0 | 1 | 1 | 0 | 0 | 0 | eMIOS_0 channel 16     |
| 0         | 0 | 1 | 1 | 0 | 0 | 1 | eMIOS_0 channel 17     |
| 0         | 0 | 1 | 1 | 0 | 1 | 0 | eMIOS_0 channel 18     |
| 0         | 0 | 1 | 1 | 0 | 1 | 1 | eMIOS_0 channel 19     |
| 0         | 0 | 1 | 1 | 1 | 0 | 0 | eMIOS_0 channel 20     |
| 0         | 0 | 1 | 1 | 1 | 0 | 1 | eMIOS_0 channel 21     |
| 0         | 0 | 1 | 1 | 1 | 1 | 0 | eMIOS_0 channel 22     |
| 0         | 0 | 1 | 1 | 1 | 1 | 1 | eMIOS_0 channel 23     |
| 0         | 1 | x | x | x | x | x | eTPUA30                |
| 1         | 0 | x | x | x | x | x | eMIOS_1 channel 2      |
| 1         | 1 | x | x | x | x | x | ETRIG0 Pin             |

**Table 8-1050. eQADC\_B Command FIFO 1 Trigger Sources**

| cTSEL_1_B |   |   |   |   |   |   | eQADC_B Trigger Inputs  |
|-----------|---|---|---|---|---|---|-------------------------|
| 0         | 0 | 0 | 0 | 0 | 0 | 0 | Not Connected (default) |
| 0         | 0 | 0 | 0 | 0 | 0 | 1 | RTI Trigger             |
| 0         | 0 | 0 | 0 | 0 | 1 | 0 | PIT0 Trigger            |
| 0         | 0 | 0 | 0 | 0 | 1 | 1 | PIT1 Trigger            |
| 0         | 0 | 0 | 0 | 1 | 0 | 0 | PIT2 Trigger            |
| 0         | 0 | 0 | 0 | 1 | 0 | 1 | PIT3 Trigger            |
| 0         | 0 | 0 | 0 | 1 | 1 | 0 | eTPUA7                  |
| 0         | 0 | 0 | 0 | 1 | 1 | 1 | ETRIG0 pin              |

Table continues on the next page...

Table 8-1050. eQADC\_B Command FIFO 1 Trigger Sources (continued)

| cTSEL_1_B |   |   |   |   |   |   | eQADC_B Trigger Inputs |
|-----------|---|---|---|---|---|---|------------------------|
| 0         | 0 | 0 | 1 | 0 | x | x | Reserved               |
| 0         | 0 | 0 | 1 | 1 | 0 | 0 | eTPUA28                |
| 0         | 0 | 0 | 1 | 1 | 0 | 1 | eTPUA29                |
| 0         | 0 | 0 | 1 | 1 | 1 | 0 | eTPUA30                |
| 0         | 0 | 0 | 1 | 1 | 1 | 1 | eTPUA31                |
| 0         | 0 | 1 | 0 | 0 | 0 | 0 | eTPUB28                |
| 0         | 0 | 1 | 0 | 0 | 0 | 1 | eTPUB29                |
| 0         | 0 | 1 | 0 | 0 | 1 | 0 | eTPUB30                |
| 0         | 0 | 1 | 0 | 0 | 1 | 1 | eTPUB31                |
| 0         | 0 | 1 | 0 | 1 | 0 | 0 | eTPUC28                |
| 0         | 0 | 1 | 0 | 1 | 0 | 1 | eTPUC29                |
| 0         | 0 | 1 | 0 | 1 | 1 | 0 | eTPUC30                |
| 0         | 0 | 1 | 0 | 1 | 1 | 1 | eTPUC31                |
| 0         | 0 | 1 | 1 | 0 | 0 | 0 | eMIOS_0 channel 16     |
| 0         | 0 | 1 | 1 | 0 | 0 | 1 | eMIOS_0 channel 17     |
| 0         | 0 | 1 | 1 | 0 | 1 | 0 | eMIOS_0 channel 18     |
| 0         | 0 | 1 | 1 | 0 | 1 | 1 | eMIOS_0 channel 19     |
| 0         | 0 | 1 | 1 | 1 | 0 | 0 | eMIOS_0 channel 20     |
| 0         | 0 | 1 | 1 | 1 | 0 | 1 | eMIOS_0 channel 21     |
| 0         | 0 | 1 | 1 | 1 | 1 | 0 | eMIOS_0 channel 22     |
| 0         | 0 | 1 | 1 | 1 | 1 | 1 | eMIOS_0 channel 23     |
| 0         | 1 | x | x | x | x | x | eTPUA31                |
| 1         | 0 | x | x | x | x | x | eMIOS_1 channel 3      |
| 1         | 1 | x | x | x | x | x | ETRIG1 Pin             |

Address: 0h base + 91Ch offset = 91Ch

|       |    |          |    |    |    |    |    |    |    |          |    |    |    |    |    |    |
|-------|----|----------|----|----|----|----|----|----|----|----------|----|----|----|----|----|----|
| Bit   | 0  | 1        | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9        | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0  | CTSEL1_B |    |    |    |    |    |    | 0  | CTSEL0_B |    |    |    |    |    |    |
| W     |    |          |    |    |    |    |    |    |    |          |    |    |    |    |    |    |
| Reset | 0  | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0        | 0  | 0  | 0  | 0  | 0  | 0  |
| Bit   | 16 | 17       | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25       | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0  |          |    |    |    |    |    |    |    |          |    |    |    |    |    |    |
| W     |    |          |    |    |    |    |    |    |    |          |    |    |    |    |    |    |
| Reset | 0  | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0        | 0  | 0  | 0  | 0  | 0  | 0  |

## SIU\_ISEL7 field descriptions

| Field         | Description                                                                             |
|---------------|-----------------------------------------------------------------------------------------|
| 0<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |

Table continues on the next page...

**SIU\_ISEL7 field descriptions (continued)**

| Field             | Description                                                                             |
|-------------------|-----------------------------------------------------------------------------------------|
| 1–7<br>CTSEL1_B   | Trigger source selection for EQADC_B Command FIFO 1.                                    |
| 8<br>Reserved     | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 9–15<br>CTSEL0_B  | Trigger source selection for EQADC_B Command FIFO 0.                                    |
| 16–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |

**8.2.22 eTPU\_A Input Select Register (SIU\_ISEL8)**

The SIU\_ISEL8 register is used to multiplex the eTPU\_A[24:29] inputs. These 6 eTPU\_A channels can come from the output of the DSPI or the corresponding pad. When SIU\_ISEL8 is in its default state, the eTPU pins listed in the figure below will not be enabled as inputs, irrespective of the SIU\_PCR[PA] field (PCR138-143).

Address: 0h base + 920h offset = 920h

|       |    |    |    |        |    |    |    |        |    |    |    |        |    |    |    |        |
|-------|----|----|----|--------|----|----|----|--------|----|----|----|--------|----|----|----|--------|
| Bit   | 0  | 1  | 2  | 3      | 4  | 5  | 6  | 7      | 8  | 9  | 10 | 11     | 12 | 13 | 14 | 15     |
| R     | 0  |    |    |        |    |    |    |        |    |    |    | ETPU29 | 0  |    |    | ETPU28 |
| W     |    |    |    |        |    |    |    |        |    |    |    |        |    |    |    |        |
| Reset | 0  | 0  | 0  | 0      | 0  | 0  | 0  | 0      | 0  | 0  | 0  | 0      | 0  | 0  | 0  | 0      |
| Bit   | 16 | 17 | 18 | 19     | 20 | 21 | 22 | 23     | 24 | 25 | 26 | 27     | 28 | 29 | 30 | 31     |
| R     | 0  |    |    | ETPU27 | 0  |    |    | ETPU26 | 0  |    |    | ETPU25 | 0  |    |    | ETPU24 |
| W     |    |    |    |        |    |    |    |        |    |    |    |        |    |    |    |        |
| Reset | 0  | 0  | 0  | 0      | 0  | 0  | 0  | 0      | 0  | 0  | 0  | 0      | 0  | 0  | 0  | 0      |

**SIU\_ISEL8 field descriptions**

| Field            | Description                                                                                                                                                               |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–10<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                   |
| 11<br>ETPU29     | eTPU29 input select. Specifies the source of the eTPU29 channel input.<br>0 DSPI_B Serialized input 8.<br>1 eTPUA29 pin or ANB10 pin, as selected by SIU_IMUX7[MUXSEL13]. |

Table continues on the next page...

**SIU\_ISEL8 field descriptions (continued)**

| Field             | Description                                                                                                                                                                |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 12–14<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                    |
| 15<br>ETPU28      | eTPU28 input select. Specifies the source of the eTPU28 channel input.<br>0 DSPI_B Serialized input 9.<br>1 eTPUA28 pin or ANB11 pin, as selected by SIU_IMUX7[MUXSEL12].  |
| 16–18<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                    |
| 19<br>ETPU27      | eTPU27 input select. Specifies the source of the eTPU27 channel input.<br>0 DSPI_B Serialized input 10.<br>1 eTPUA27 pin or ANB12 pin, as selected by SIU_IMUX7[MUXSEL11]. |
| 20–22<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                    |
| 23<br>ETPU26      | eTPU26 input select. Specifies the source of the eTPU26 channel input.<br>0 DSPI_B Serialized input 11.<br>1 eTPUA26 pin or ANB13 pin, as selected by SIU_IMUX7[MUXSEL10]. |
| 24–26<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                    |
| 27<br>ETPU25      | eTPU25 input select. Specifies the source of the eTPU25 channel input.<br>0 DSPI_B Serialized input 12.<br>1 eTPUA25 pin or ANB14 pin, as selected by SIU_IMUX7[MUXSEL9].  |
| 28–30<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                    |
| 31<br>ETPU24      | eTPU24 input select. Specifies the source of the eTPU24 channel input.<br>0 DSPI_B Serialized input 13.<br>1 eTPUA24 pin or ANB15 pin, as selected by SIU_IMUX7[MUXSEL8].  |

**8.2.23 eQADC Advance Trigger Source Register (SIU\_ISEL9)**

The eQADC streaming mode requires a second trigger for command queue 0. Selectable source options for this trigger consist of eTPU, eMIOS or PIT channels. This mux register selects the source of the command queue 0 trigger.

**Table 8-1053. eQADC\_A CFIFO0 Advance Trigger Source Selection**

| eETSEL0ADV_A |   |   |   |   | eQADC Trigger Source |
|--------------|---|---|---|---|----------------------|
| 0            | 0 | 0 | 0 | 0 | Reserved             |
| 0            | 0 | 0 | 0 | 1 | RTI Trigger          |
| 0            | 0 | 0 | 1 | 0 | PIT0 Trigger         |
| 0            | 0 | 0 | 1 | 1 | PIT1 Trigger         |

*Table continues on the next page...*

**Table 8-1053. eQADC\_A CFIFO0 Advance Trigger Source Selection  
(continued)**

| eETSEL0ADV_A |   |   |   |   | eQADC Trigger Source       |
|--------------|---|---|---|---|----------------------------|
| 0            | 0 | 1 | 0 | 0 | PIT2 Trigger               |
| 0            | 0 | 1 | 0 | 1 | PIT3 Trigger               |
| 0            | 0 | 1 | 1 | 0 | Reserved                   |
| 0            | 0 | 1 | 1 | 1 | Reserved                   |
| 0            | 1 | 0 | 0 | 0 | eTPUA30 AND PIT0           |
| 0            | 1 | 0 | 0 | 1 | eTPUA30 AND PIT1           |
| 0            | 1 | 0 | 1 | 0 | Reserved                   |
| 0            | 1 | 0 | 1 | 1 | Reserved                   |
| 0            | 1 | 1 | 0 | 0 | eTPUA28                    |
| 0            | 1 | 1 | 0 | 1 | eTPUA29                    |
| 0            | 1 | 1 | 1 | 0 | eTPUA30                    |
| 0            | 1 | 1 | 1 | 1 | eTPUA31                    |
| 1            | 0 | 0 | 0 | 0 | eTPUC28                    |
| 1            | 0 | 0 | 0 | 1 | eTPUC29                    |
| 1            | 0 | 0 | 1 | 0 | eTPUC30                    |
| 1            | 0 | 0 | 1 | 1 | eTPUC31                    |
| 1            | 0 | 1 | 0 | 0 | eMIOS_1 channel 2 AND PIT2 |
| 1            | 0 | 1 | 0 | 1 | eMIOS_1 channel 2 AND PIT3 |
| 1            | 0 | 1 | 1 | 0 | Reserved                   |
| 1            | 0 | 1 | 1 | 1 | Reserved                   |
| 1            | 1 | 0 | 0 | 0 | Reserved                   |
| 1            | 1 | 0 | 0 | 1 | Reserved                   |
| 1            | 1 | 0 | 1 | 0 | Reserved                   |
| 1            | 1 | 0 | 1 | 1 | Reserved                   |
| 1            | 1 | 1 | 0 | 0 | Reserved                   |
| 1            | 1 | 1 | 0 | 1 | Reserved                   |
| 1            | 1 | 1 | 1 | 0 | Reserved                   |
| 1            | 1 | 1 | 1 | 1 | eMIOS_0 channel 23         |

**Table 8-1054. eQADC\_B CFIFO0 Advance Trigger Source Selection**

| eETSEL0ADV_B |   |   |   |   | eQADC Trigger Source |
|--------------|---|---|---|---|----------------------|
| 0            | 0 | 0 | 0 | 0 | Reserved             |
| 0            | 0 | 0 | 0 | 1 | RTI Trigger          |
| 0            | 0 | 0 | 1 | 0 | PIT0 Trigger         |
| 0            | 0 | 0 | 1 | 1 | PIT1 Trigger         |
| 0            | 0 | 1 | 0 | 0 | PIT2 Trigger         |
| 0            | 0 | 1 | 0 | 1 | PIT3 Trigger         |

Table continues on the next page...

**Table 8-1054. eQADC\_B CFIFO0 Advance Trigger Source Selection  
(continued)**

| eETSEL0ADV_B |   |   |   |   | eQADC Trigger Source       |
|--------------|---|---|---|---|----------------------------|
| 0            | 0 | 1 | 1 | 0 | Reserved                   |
| 0            | 0 | 1 | 1 | 1 | Reserved                   |
| 0            | 1 | 0 | 0 | 0 | eTPUA30 AND PIT0           |
| 0            | 1 | 0 | 0 | 1 | eTPUA30 AND PIT1           |
| 0            | 1 | 0 | 1 | 0 | Reserved                   |
| 0            | 1 | 0 | 1 | 1 | Reserved                   |
| 0            | 1 | 1 | 0 | 0 | eTPUA28                    |
| 0            | 1 | 1 | 0 | 1 | eTPUA29                    |
| 0            | 1 | 1 | 1 | 0 | eTPUA30                    |
| 0            | 1 | 1 | 1 | 1 | eTPUA31                    |
| 1            | 0 | 0 | 0 | 0 | eTPUC28                    |
| 1            | 0 | 0 | 0 | 1 | eTPUC29                    |
| 1            | 0 | 0 | 1 | 0 | eTPUC30                    |
| 1            | 0 | 0 | 1 | 1 | eTPUC31                    |
| 1            | 0 | 1 | 0 | 0 | eMIOS_1 channel 2 AND PIT2 |
| 1            | 0 | 1 | 0 | 1 | eMIOS_1 channel 2 AND PIT3 |
| 1            | 0 | 1 | 1 | 0 | Reserved                   |
| 1            | 0 | 1 | 1 | 1 | Reserved                   |
| 1            | 1 | 0 | 0 | 0 | Reserved                   |
| 1            | 1 | 0 | 0 | 1 | Reserved                   |
| 1            | 1 | 0 | 1 | 0 | Reserved                   |
| 1            | 1 | 0 | 1 | 1 | Reserved                   |
| 1            | 1 | 1 | 0 | 0 | Reserved                   |
| 1            | 1 | 1 | 0 | 1 | Reserved                   |
| 1            | 1 | 1 | 1 | 0 | Reserved                   |
| 1            | 1 | 1 | 1 | 1 | Reserved                   |
| 1            | 1 | 1 | 1 | 1 | eMIOS_0 channel 23         |

Address: 0h base + 924h offset = 924h

|       |   |   |   |   |   |   |   |   |   |   |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |              |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------------|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10           | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25           | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   | EETSEL0ADV_A |    |    |    |    | 0  |    |    |    |    |    |    |    |    |    | EETSEL0ADV_B |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |              |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0            | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0            | 0  | 0  | 0  | 0  | 0  | 0  |

### SIU\_ISEL9 field descriptions

| Field                 | Description                                                                             |
|-----------------------|-----------------------------------------------------------------------------------------|
| 0–10<br>Reserved      | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 11–15<br>EETSEL0ADV_A | EQADC_A CFIFO0 Advance Trigger Source Selection                                         |

Table continues on the next page...



**SIU\_ISEL9 field descriptions (continued)**

| Field                 | Description                                                                             |
|-----------------------|-----------------------------------------------------------------------------------------|
| 16–26<br>Reserved     | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 27–31<br>EETSEL0ADV_B | EQADC_B CFIFO0 Advance Trigger Source Selection                                         |

**8.2.24 Sigma-Delta Gate Selection Register (SIU\_SDGATE\_SEL)**

Each Sigma-Delta ADC supports an input that gates (disable) interrupt and DMA requests. The SIU\_SDGATE\_SEL register selects the source of the interrupt/DMA gating input. The sources selected by this register come from eTPU output channels. Gating DMA and interrupt requests is useful for applications like knock processing when —outside the knock window—the ADC results are not needed, but the Sigma-Delta ADC is still converting due to its architecture.

Address: 0h base + 940h offset = 940h

|       |               |    |    |    |               |    |    |    |               |    |    |    |               |    |    |    |
|-------|---------------|----|----|----|---------------|----|----|----|---------------|----|----|----|---------------|----|----|----|
| Bit   | 0             | 1  | 2  | 3  | 4             | 5  | 6  | 7  | 8             | 9  | 10 | 11 | 12            | 13 | 14 | 15 |
| R     | 0             |    |    |    |               |    |    |    |               |    |    |    |               |    |    |    |
| W     |               |    |    |    |               |    |    |    |               |    |    |    |               |    |    |    |
| Reset | 0             | 0  | 0  | 0  | 0             | 0  | 0  | 0  | 0             | 0  | 0  | 0  | 0             | 0  | 0  | 0  |
| Bit   | 16            | 17 | 18 | 19 | 20            | 21 | 22 | 23 | 24            | 25 | 26 | 27 | 28            | 29 | 30 | 31 |
| R     | SD_D_GATE_SEL |    |    |    | SD_C_GATE_SEL |    |    |    | SD_B_GATE_SEL |    |    |    | SD_A_GATE_SEL |    |    |    |
| W     |               |    |    |    |               |    |    |    |               |    |    |    |               |    |    |    |
| Reset | 0             | 0  | 0  | 0  | 0             | 0  | 0  | 0  | 0             | 0  | 0  | 0  | 0             | 0  | 0  | 0  |

**SIU\_SDGATE\_SEL field descriptions**

| Field                  | Description                                                                                                                                                                                                                       |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>Reserved       | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                           |
| 16–19<br>SD_D_GATE_SEL | DMA/Interrupt gating input selection for Sigma-Delta ADC D.<br><br>0000 eTPU_A Channel 12.<br>0001 eTPU_A Channel 30.<br>0010 eTPU_B Channel 12.<br>0011 eTPU_B Channel 30.<br>0100 eTPU_C Channel 10.<br>0101 eTPU_C Channel 30. |
| 20–23<br>SD_C_GATE_SEL | DMA/Interrupt gating input selection for Sigma-Delta ADC C.<br><br>0000 eTPU_A Channel 11.<br>0001 eTPU_A Channel 19.<br>0010 eTPU_B Channel 11.                                                                                  |

*Table continues on the next page...*

## SIU\_SDGATE\_SEL field descriptions (continued)

| Field                  | Description                                                                                                                                                                                                                    |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                        | 0011 eTPU_B Channel 19.<br>0100 eTPU_C Channel 9.<br>0101 eTPU_C Channel 29.                                                                                                                                                   |
| 24–27<br>SD_B_GATE_SEL | DMA/Interrupt gating input selection for Sigma-Delta ADC B.<br><br>0000 eTPU_A Channel 10.<br>0001 eTPU_A Channel 2.<br>0010 eTPU_B Channel 10.<br>0011 eTPU_B Channel 2.<br>0100 eTPU_C Channel 8.<br>0101 eTPU_C Channel 28. |
| 28–31<br>SD_A_GATE_SEL | DMA/Interrupt gating input selection for Sigma-Delta ADC A.<br><br>0000 eTPU_A Channel 9.<br>0001 eTPU_A Channel 1.<br>0010 eTPU_B Channel 9.<br>0011 eTPU_B Channel 1.<br>0100 eTPU_C Channel 7.<br>0101 eTPU_C Channel 27.   |

## 8.2.25 Sigma-Delta to eTPU\_A Connection Configuration Register (SIU\_SDETPUA)

This register configures the connection between Sigma-Delta ADCs and eTPU\_A input channels. Two ADC signals can be connected to eTPU channel inputs. The first one is a signal that indicates valid data conversion completion. The second one is a watchdog signal that flags a conversion which is outside pre-programmed high/low threshold limits.

Address: 0h base + 944h offset = 944h

|       |      |      |     |      |     |     |     |     |    |     |      |      |      |    |      |      |
|-------|------|------|-----|------|-----|-----|-----|-----|----|-----|------|------|------|----|------|------|
| Bit   | 0    | 1    | 2   | 3    | 4   | 5   | 6   | 7   | 8  | 9   | 10   | 11   | 12   | 13 | 14   | 15   |
| R     | 0    |      |     | 0    |     |     |     |     | 0  |     |      |      |      | 0  |      |      |
| W     |      | CH1  | CH2 |      | CH4 | CH5 | CH6 | CH7 |    | CH9 | CH10 | CH11 | CH12 |    | CH14 | CH15 |
| Reset | 0    | 0    | 0   | 0    | 0   | 0   | 0   | 0   | 0  | 0   | 0    | 0    | 0    | 0  | 0    | 0    |
| Bit   | 16   | 17   | 18  | 19   | 20  | 21  | 22  | 23  | 24 | 25  | 26   | 27   | 28   | 29 | 30   | 31   |
| R     |      |      | 0   |      | 0   |     |     |     |    |     |      |      |      |    |      | 0    |
| W     | CH16 | CH17 |     | CH19 |     |     |     |     |    |     |      |      |      |    | CH30 |      |
| Reset | 0    | 0    | 0   | 0    | 0   | 0   | 0   | 0   | 0  | 0   | 0    | 0    | 0    | 0  | 0    | 0    |

**SIU\_SDETPUA field descriptions**

| Field          | Description                                                                                                                                                                          |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                              |
| 1<br>CH1       | eTPU_A channel 1 input select. Specifies the source of the eTPU_A channel 1 input.<br><br>0 eTPU_A channel 1 input pad.<br>1 Sigma-Delta A conversion limit detection watchdog flag. |
| 2<br>CH2       | eTPU_A channel 2 input select. Specifies the source of the eTPU_A channel 2 input.<br><br>0 eTPU_A channel 2 input pad.<br>1 Sigma-Delta B conversion limit detection watchdog flag. |
| 3<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                              |
| 4<br>CH4       | eTPU_A channel 4 input select. Specifies the source of the eTPU_A channel 4 input.<br><br>0 eTPU_A channel 4 input pad.<br>1 Sigma-Delta A conversion limit detection watchdog flag. |
| 5<br>CH5       | eTPU_A channel 5 input select. Specifies the source of the eTPU_A channel 5 input.<br><br>0 eTPU_A channel 5 input pad.<br>1 Sigma-Delta B conversion limit detection watchdog flag. |
| 6<br>CH6       | eTPU_A channel 6 input select. Specifies the source of the eTPU_A channel 6 input.<br><br>0 eTPU_A channel 6 input pad.<br>1 Sigma-Delta C conversion limit detection watchdog flag. |
| 7<br>CH7       | eTPU_A channel 7 input select. Specifies the source of the eTPU_A channel 7 input.<br><br>0 eTPU_A channel 7 input pad.<br>1 Sigma-Delta D conversion limit detection watchdog flag. |
| 8<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                              |
| 9<br>CH9       | eTPU_A channel 9 input select. Specifies the source of the eTPU_A channel 9 input.<br><br>0 eTPU_A channel 9 input pad.<br>1 Sigma-Delta A data valid output signal.                 |
| 10<br>CH10     | eTPU_A channel 10 input select. Specifies the source of the eTPU_A channel 10 input.<br><br>0 eTPU_A channel 10 input pad.<br>1 Sigma-Delta B data valid output signal.              |
| 11<br>CH11     | eTPU_A channel 11 input select. Specifies the source of the eTPU_A channel 11 input.<br><br>0 eTPU_A channel 11 input pad.<br>1 Sigma-Delta C data valid output signal.              |
| 12<br>CH12     | eTPU_A channel 12 input select. Specifies the source of the eTPU_A channel 12 input.<br><br>0 eTPU_A channel 12 input pad.<br>1 Sigma-Delta D data valid output signal.              |
| 13<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                              |

*Table continues on the next page...*

**SIU\_SDETPUA field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                               |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 14<br>CH14        | eTPU_A channel 14 input select. Specifies the source of the eTPU_A channel 14 input.<br>0 eTPU_A channel 14 input pad.<br>1 Sigma-Delta A data valid output signal.                                                                       |
| 15<br>CH15        | eTPU_A channel 15 input select. Specifies the source of the eTPU_A channel 15 input.<br>0 eTPU_A channel 15 input pad.<br>1 Sigma-Delta B data valid output signal.                                                                       |
| 16<br>CH16        | eTPU_A channel 16 input select. Specifies the source of the eTPU_A channel 16 input.<br>0 eTPU_A channel 16 input pad.<br>1 Sigma-Delta C data valid output signal.                                                                       |
| 17<br>CH17        | eTPU_A channel 17 input select. Specifies the source of the eTPU_A channel 17 input.<br>0 eTPU_A channel 17 input pad.<br>1 Sigma-Delta D data valid output signal.                                                                       |
| 18<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                   |
| 19<br>CH19        | eTPU_A channel 19 input select. Specifies the source of the eTPU_A channel 19 input.<br>0 eTPU_A channel 19 input pad.<br>1 Sigma-Delta C conversion limit detection watchdog flag.                                                       |
| 20–29<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                   |
| 30<br>CH30        | eTPU_A channel 30 input select. Specifies the source of the eTPU_A channel 30 input.<br>0 Output from multiplex controlled by SIU_IMUX7[MUXSEL14] (eTPUA30 pin or ANB9 pin).<br>1 Sigma-Delta D conversion limit detection watchdog flag. |
| 31<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                   |

## 8.2.26 Sigma-Delta to eTPU\_B Connection Configuration Register (SIU\_SDETPUB)

This register configures the connection between Sigma-Delta ADCs and eTPU\_B input channels. Two ADC signals can be connected to eTPU channel inputs. The first one is a signal that indicates valid data conversion completion. The second one is a watchdog signal that flags a conversion which is outside pre-programmed high/low threshold limits.

Address: 0h base + 948h offset = 948h

| Bit   | 0 | 1   | 2   | 3 | 4   | 5   | 6   | 7   | 8 | 9   | 10   | 11   | 12   | 13 | 14   | 15   |
|-------|---|-----|-----|---|-----|-----|-----|-----|---|-----|------|------|------|----|------|------|
| R     | 0 |     |     | 0 |     |     |     |     | 0 |     |      |      |      | 0  |      |      |
| W     |   | CH1 | CH2 |   | CH4 | CH5 | CH6 | CH7 |   | CH9 | CH10 | CH11 | CH12 |    | CH14 | CH15 |
| Reset | 0 | 0   | 0   | 0 | 0   | 0   | 0   | 0   | 0 | 0   | 0    | 0    | 0    | 0  | 0    | 0    |

| Bit   | 16   | 17   | 18 | 19   | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30   | 31 |
|-------|------|------|----|------|----|----|----|----|----|----|----|----|----|----|------|----|
| R     |      |      | 0  |      | 0  |    |    |    |    |    |    |    |    |    |      |    |
| W     | CH16 | CH17 |    | CH19 |    |    |    |    |    |    |    |    |    |    | CH30 |    |
| Reset | 0    | 0    | 0  | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0  |

**SIU\_SDETPUB field descriptions**

| Field         | Description                                                                                                                                                                      |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                          |
| 1<br>CH1      | eTPU_B channel 1 input select. Specifies the source of the eTPU_B channel 1 input.<br>0 eTPU_B channel 1 input pad.<br>1 Sigma-Delta A conversion limit detection watchdog flag. |
| 2<br>CH2      | eTPU_B channel 2 input select. Specifies the source of the eTPU_B channel 2 input.<br>0 eTPU_B channel 2 input pad.<br>1 Sigma-Delta B conversion limit detection watchdog flag. |
| 3<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                          |
| 4<br>CH4      | eTPU_B channel 4 input select. Specifies the source of the eTPU_B channel 4 input.<br>0 eTPU_B channel 4 input pad.<br>1 Sigma-Delta A conversion limit detection watchdog flag. |
| 5<br>CH5      | eTPU_B channel 5 input select. Specifies the source of the eTPU_B channel 5 input.                                                                                               |

*Table continues on the next page...*

**SIU\_SDETPUB field descriptions (continued)**

| Field          | Description                                                                                                                                                                      |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | 0 eTPU_B channel 5 input pad.<br>1 Sigma-Delta B conversion limit detection watchdog flag.                                                                                       |
| 6<br>CH6       | eTPU_B channel 6 input select. Specifies the source of the eTPU_B channel 6 input.<br>0 eTPU_B channel 6 input pad.<br>1 Sigma-Delta C conversion limit detection watchdog flag. |
| 7<br>CH7       | eTPU_B channel 7 input select. Specifies the source of the eTPU_B channel 7 input.<br>0 eTPU_B channel 7 input pad.<br>1 Sigma-Delta D conversion limit detection watchdog flag. |
| 8<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                          |
| 9<br>CH9       | eTPU_B channel 9 input select. Specifies the source of the eTPU_B channel 9 input.<br>0 eTPU_B channel 9 input pad.<br>1 Sigma-Delta A data valid output signal.                 |
| 10<br>CH10     | eTPU_B channel 10 input select. Specifies the source of the eTPU_B channel 10 input.<br>0 eTPU_B channel 10 input pad.<br>1 Sigma-Delta B data valid output signal.              |
| 11<br>CH11     | eTPU_B channel 11 input select. Specifies the source of the eTPU_B channel 11 input.<br>0 eTPU_B channel 11 input pad.<br>1 Sigma-Delta C data valid output signal.              |
| 12<br>CH12     | eTPU_B channel 12 input select. Specifies the source of the eTPU_B channel 12 input.<br>0 eTPU_B channel 12 input pad.<br>1 Sigma-Delta D data valid output signal.              |
| 13<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                          |
| 14<br>CH14     | eTPU_B channel 14 input select. Specifies the source of the eTPU_B channel 14 input.<br>0 eTPU_B channel 14 input pad.<br>1 Sigma-Delta A data valid output signal.              |
| 15<br>CH15     | eTPU_B channel 15 input select. Specifies the source of the eTPU_B channel 15 input.<br>0 eTPU_B channel 15 input pad.<br>1 Sigma-Delta B data valid output signal.              |
| 16<br>CH16     | eTPU_B channel 16 input select. Specifies the source of the eTPU_B channel 16 input.<br>0 eTPU_B channel 16 input pad.<br>1 Sigma-Delta C data valid output signal.              |
| 17<br>CH17     | eTPU_B channel 17 input select. Specifies the source of the eTPU_B channel 17 input.<br>0 eTPU_B channel 17 input pad.<br>1 Sigma-Delta D data valid output signal.              |
| 18<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                          |

*Table continues on the next page...*

**SIU\_SDETPUB field descriptions (continued)**

| Field             | Description                                                                                                                                                                         |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 19<br>CH19        | eTPU_B channel 19 input select. Specifies the source of the eTPU_B channel 19 input.<br>0 eTPU_B channel 19 input pad.<br>1 Sigma-Delta C conversion limit detection watchdog flag. |
| 20–29<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                             |
| 30<br>CH30        | eTPU_B channel 30 input select. Specifies the source of the eTPU_B channel 30 input.<br>0 eTPU_B channel 30 input pad.<br>1 Sigma-Delta D conversion limit detection watchdog flag. |
| 31<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                             |

**8.2.27 Sigma-Delta to eTPU\_C Connection Configuration Register (SIU\_SDETPUC)**

This register configures the connection between Sigma-Delta ADCs and eTPU\_C input channels. Two ADC signals can be connected to eTPU channel inputs. The first one is a signal that indicates valid data conversion completion. The second one is a watchdog signal that flags a conversion which is outside pre-programmed high/low threshold limits.

Address: 0h base + 94Ch offset = 94Ch

|       |   |   |     |     |     |     |   |     |     |     |      |    |      |      |      |    |
|-------|---|---|-----|-----|-----|-----|---|-----|-----|-----|------|----|------|------|------|----|
| Bit   | 0 | 1 | 2   | 3   | 4   | 5   | 6 | 7   | 8   | 9   | 10   | 11 | 12   | 13   | 14   | 15 |
| R     | 0 |   |     |     |     |     | 0 |     |     |     | CH10 | 0  | CH12 | CH13 | CH14 | 0  |
| W     |   |   | CH2 | CH3 | CH4 | CH5 |   | CH7 | CH8 | CH9 |      |    | CH12 | CH13 | CH14 |    |
| Reset | 0 | 0 | 0   | 0   | 0   | 0   | 0 | 0   | 0   | 0   | 0    | 0  | 0    | 0    | 0    | 0  |

|       |    |    |    |    |    |    |    |    |    |      |    |      |      |      |      |    |
|-------|----|----|----|----|----|----|----|----|----|------|----|------|------|------|------|----|
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25   | 26 | 27   | 28   | 29   | 30   | 31 |
| R     | 0  |    |    |    |    |    |    |    |    | CH25 | 0  | CH27 | CH28 | CH29 | CH30 | 0  |
| W     |    |    |    |    |    |    |    |    |    | CH25 |    | CH27 | CH28 | CH29 | CH30 |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0  | 0    | 0    | 0    | 0    | 0  |

**SIU\_SDETPUC field descriptions**

| Field           | Description                                                                             |
|-----------------|-----------------------------------------------------------------------------------------|
| 0–1<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |

*Table continues on the next page...*

**SIU\_SDETPUC field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                               |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2<br>CH2       | eTPU_C channel 2 input select. Specifies the source of the eTPU_C channel 2 input.<br>0 eTPU_C channel 2 input pad.<br>1 Sigma-Delta A conversion limit detection watchdog flag.                                                          |
| 3<br>CH3       | eTPU_C channel 3 input select. Specifies the source of the eTPU_C channel 3 input.<br>0 Output from multiplex controlled by SIU_IMUX10[MUXSEL3] (eTPUC3 pin or PLLCFG2 pin).<br>1 Sigma-Delta B conversion limit detection watchdog flag. |
| 4<br>CH4       | eTPU_C channel 4 input select. Specifies the source of the eTPU_C channel 4 input.<br>0 eTPU_C channel 4 input pad.<br>1 Sigma-Delta C conversion limit detection watchdog flag.                                                          |
| 5<br>CH5       | eTPU_C channel 5 input select. Specifies the source of the eTPU_C channel 5 input.<br>0 eTPU_C channel 5 input pad.<br>1 Sigma-Delta D conversion limit detection watchdog flag.                                                          |
| 6<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                   |
| 7<br>CH7       | eTPU_C channel 7 input select. Specifies the source of the eTPU_C channel 7 input.<br>0 eTPU_C channel 7 input pad.<br>1 Sigma-Delta A data valid output signal.                                                                          |
| 8<br>CH8       | eTPU_C channel 8 input select. Specifies the source of the eTPU_C channel 8 input.<br>0 eTPUC8 pin or ANB23 pin, as selected by SIU_IMUX10[MUXSEL8].<br>1 Sigma-Delta B data valid output signal.                                         |
| 9<br>CH9       | eTPU_C channel 9 input select. Specifies the source of the eTPU_C channel 9 input.<br>0 eTPUC9 pin or ANB22 pin, as selected by SIU_IMUX10[MUXSEL9].<br>1 Sigma-Delta C data valid output signal.                                         |
| 10<br>CH10     | eTPU_C channel 10 input select. Specifies the source of the eTPU_C channel 10 input.<br>0 eTPUC10 pin or ANB21 pin, as selected by SIU_IMUX10[MUXSEL10].<br>1 Sigma-Delta D data valid output signal.                                     |
| 11<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                   |
| 12<br>CH12     | eTPU_C channel 12 input select. Specifies the source of the eTPU_C channel 12 input.<br>0 eTPUC12 pin or ANB19 pin, as selected by SIU_IMUX10[MUXSEL12].<br>1 Sigma-Delta A data valid output signal.                                     |
| 13<br>CH13     | eTPU_C channel 13 input select. Specifies the source of the eTPU_C channel 13 input.<br>0 eTPUC13 pin or ANB18 pin, as selected by SIU_IMUX10[MUXSEL13].<br>1 Sigma-Delta B data valid output signal.                                     |
| 14<br>CH14     | eTPU_C channel 14 input select. Specifies the source of the eTPU_C channel 14 input.<br>0 eTPUC14 pin or ANB17 pin, as selected by SIU_IMUX10[MUXSEL14].<br>1 Sigma-Delta C data valid output signal.                                     |

*Table continues on the next page...*



**SIU\_SDETPUC field descriptions (continued)**

| Field             | Description                                                                                                                                                                             |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 15–24<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                 |
| 25<br>CH25        | eTPU_C channel 25 input select. Specifies the source of the eTPU_C channel 25 input.<br><br>0 eTPU_C channel 25 input pad.<br>1 Sigma-Delta D data valid output signal.                 |
| 26<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                 |
| 27<br>CH27        | eTPU_C channel 27 input select. Specifies the source of the eTPU_C channel 27 input.<br><br>0 eTPU_C channel 27 input pad.<br>1 Sigma-Delta A conversion limit detection watchdog flag. |
| 28<br>CH28        | eTPU_C channel 28 input select. Specifies the source of the eTPU_C channel 28 input.<br><br>0 eTPU_C channel 28 input pad.<br>1 Sigma-Delta B conversion limit detection watchdog flag. |
| 29<br>CH29        | eTPU_C channel 29 input select. Specifies the source of the eTPU_C channel 29 input.<br><br>0 eTPU_C channel 29 input pad.<br>1 Sigma-Delta C conversion limit detection watchdog flag. |
| 30<br>CH30        | eTPU_C channel 30 input select. Specifies the source of the eTPU_C channel 30 input.<br><br>0 eTPU_C channel 30 input pad.<br>1 Sigma-Delta D conversion limit detection watchdog flag. |
| 31<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                 |

## 8.2.28 Trigger Burst Generator Control Registers for eQADC\_A (SIU\_TBГ\_CR\_An)

These registers provide control bits for the Trigger Burst Generators (TBG) connected at the trigger inputs of the 6 CFIFOS of eQADC\_A. There is one register for each eQADC\_A CFIFO input. The numbering goes from 0 to 5, n=0 corresponding to address 0x950 and CFIFO0, n=5 corresponding to address 0x964 and CFIFO5. See Trigger Burst Generators for detailed information on the TBG feature.

Address: 0h base + 950h offset + (4d × i), where i=0d to 5d

|       |     |   |   |   |        |   |   |   |         |   |    |    |    |    |    |    |
|-------|-----|---|---|---|--------|---|---|---|---------|---|----|----|----|----|----|----|
| Bit   | 0   | 1 | 2 | 3 | 4      | 5 | 6 | 7 | 8       | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | GEN | 0 |   |   | CLKSEL |   | 0 |   | TRIGPER |   |    |    |    |    |    |    |
| W     |     |   |   |   |        |   |   |   |         |   |    |    |    |    |    |    |
| Reset | 0   | 0 | 0 | 0 | 0      | 0 | 0 | 0 | 0       | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 16      | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | TRIGPER |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0       | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

### SIU\_TBГ\_CR\_An field descriptions

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>GEN        | Global enable bit. This bit enables the operation of the TBG module. If not enabled the TBG is in bypass mode, thus triggers selected by ISEL4-7 are routed directly to the CFIFO input. If enabled, then the TBG behavior depends upon the clock selection CLKSEL bits.<br><br>0 TBG is disabled, module operation is transparent regarding the triggers selected by ISEL4-7.<br>1 TBG is enabled, generating triggers according to ISEL4-7 selection and clock selection.                                                                                                  |
| 1–3<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 4–5<br>CLKSEL   | Clock Selection bits. The clock selection bits select three clocks that are used to increment the TBG internal counter: system clock, ADC0 clock, ADC1 clock and no clock. If no clock is select the internal counter remains inactive and no triggers are generated to the corresponding eQADC CFIFO.<br><br>00 No clock selected. No triggers are generated independently of ISEL4-7 trigger selection.<br>01 System clock. TBG increments at system clock rate.<br>10 ADC0 clock. TBG increments at ADC0 clock rate.<br>11 ADC1 clock. TBG increments at ADC1 clock rate. |
| 6–7<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 8–31<br>TRIGPER | Trigger Period. The Trigger period field defines the time interval from one trigger to the next in a trigger burst. The trigger period value is loaded into a TBG internal counter which performs a count down. When zero value is reached one trigger is generated for the corresponding eQADC CFIFO.                                                                                                                                                                                                                                                                       |

## 8.2.29 Trigger Burst Generator Control Registers for eQADC\_B (SIU\_TBГ\_CR\_Bn)

These registers provide control bits for the Trigger Burst Generators (TBG) connected at the trigger inputs of the 6 CFIFOS of eQADC\_B. There is one register for each eQADC\_B CFIFO input. The numbering goes from 0 to 5, n=0 corresponding to address 0x968 and CFIFO0, n=5 corresponding to address 0x97C and CFIFO5. See Trigger Burst Generators for detailed information on the TBG feature.

Address: 0h base + 968h offset + (4d × i), where i=0d to 5d

|       |     |   |   |   |        |   |   |   |         |   |    |    |    |    |    |    |
|-------|-----|---|---|---|--------|---|---|---|---------|---|----|----|----|----|----|----|
| Bit   | 0   | 1 | 2 | 3 | 4      | 5 | 6 | 7 | 8       | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | GEN | 0 |   |   | CLKSEL |   | 0 |   | TRIGPER |   |    |    |    |    |    |    |
| W     |     |   |   |   |        |   |   |   |         |   |    |    |    |    |    |    |
| Reset | 0   | 0 | 0 | 0 | 0      | 0 | 0 | 0 | 0       | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 16      | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | TRIGPER |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0       | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

### SIU\_TBГ\_CR\_Bn field descriptions

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>GEN        | Global enable bit. This bit enables the operation of the TBG module. If not enabled the TBG is in bypass mode, thus triggers selected by ISEL4-7 are routed directly to the CFIFO input. If enabled, then the TBG behavior depends upon the clock selection CLKSEL bits. It is necessary to clear this bit to change the CLKSEL and TRIGPER fields.<br><br>0 TBG is disabled, module operation is transparent regarding the triggers selected by ISEL4-7.<br>1 TBG is enabled, generating triggers according to ISEL4-7 selection and clock selection.                                                                                                                                               |
| 1–3<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 4–5<br>CLKSEL   | Clock Selection bits. The clock selection bits select three clocks that are used to increment the TBG internal counter: peripheral clock (per_clk), ADC0 clock, ADC1 clock and no clock. If no clock is select the internal counter remains inactive and no triggers are generated to the corresponding eQADC CFIFO.<br><br><b>NOTE:</b> It is necessary to clear the global enable bit GEN when changing this field.<br><br>00 No clock selected. No triggers are generated independently of ISEL4-7 trigger selection.<br>01 Peripheral clock (per_clk). TBG increments at per_clk rate.<br>10 ACD0 clock. TBG increments at ADC0 clock rate.<br>11 ADC1 clock. TBG increments at ADC1 clock rate. |
| 6–7<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 8–31<br>TRIGPER | Trigger Period. The Trigger period field defines the time interval from one trigger to the next in a trigger burst. The trigger period value is loaded into a TBG internal counter which performs a count down. When zero value is reached one trigger is generated for the corresponding eQADC CFIFO.<br><br><b>NOTE:</b> It is recommended to clear the global enable bit GEN when changing this field.                                                                                                                                                                                                                                                                                            |

## 8.2.30 External Clock Control Register (SIU\_ECCR)

The SIU\_ECCR controls the timing relationship between the system clock and the external clocks ENGCLK and CLKOUT.

Address: 0h base + 984h offset = 984h

|       |        |    |    |    |    |    |    |    |      |    |    |    |      |    |      |    |
|-------|--------|----|----|----|----|----|----|----|------|----|----|----|------|----|------|----|
| Bit   | 0      | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8    | 9  | 10 | 11 | 12   | 13 | 14   | 15 |
| R     | 0      |    |    |    |    |    |    |    |      |    |    |    |      |    |      |    |
| W     |        |    |    |    |    |    |    |    |      |    |    |    |      |    |      |    |
| Reset | 0      | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0  | 0  | 0  | 0    | 0  | 0    | 0  |
| Bit   | 16     | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24   | 25 | 26 | 27 | 28   | 29 | 30   | 31 |
| R     | ENGDIV |    |    |    |    |    |    |    | ECSS | 0  |    |    | EBTS | 0  | EBDF |    |
| W     |        |    |    |    |    |    |    |    |      |    |    |    |      |    |      |    |
| Reset | 0      | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0    | 0  | 0  | 0  | 0    | 0  | 0    | 1  |

### SIU\_ECCR field descriptions

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 16–23<br>ENGDIV   | Engineering Clock Division Factor. The ENGDIV field specifies the frequency ratio between the platform clock (plat_clk) and ENGCLK. The ENGCLK frequency is divided from the platform clock frequency according to the following equation:<br>$EngineeringClockFrequency = \frac{PlatformClockFrequency}{(ENGDIV \times 2)}$ Setting ENGDIV to 0 makes the ENGCLK frequency equal to the platform clock frequency.<br><b>NOTE:</b> Maximum ENGCLK frequency is limited by the pad performance. Refer to the Electrical Specifications in the Data Sheet document for this device. |
| 24<br>ECSS        | Engineering clock source select. The ECSS bit selects the source clock to be divided to generate the ENGCLK clock.<br><br>0 The system clock is the source of the ENGCLK.<br>1 The crystal oscillator clock is the source of ENGCLK.                                                                                                                                                                                                                                                                                                                                              |
| 25–27<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 28<br>EBTS        | External Bus Tap Select. Changes the phase relationship between the system clock and CLKOUT. Changing the phase relationship so that CLKOUT is advanced in relation to system clock increases the output hold time of the external bus signals to a non-zero value. It also increases the output delay times, increases the input hold times to non-zero values, and decreases the input setup times. Refer to the Data Sheet document for how the EBTS bit affects the electrical specifications.                                                                                |

Table continues on the next page...

**SIU\_ECCR field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                                      |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | <b>NOTE:</b> The EBTS bit must not be modified while an external bus transaction is in progress.<br><br>0 External bus signals have zero output hold times.<br>1 External bus signals have non-zero output hold times.                                                           |
| 29<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                          |
| 30–31<br>EBDF  | External Bus Division Factor. Specifies the frequency ratio between the system clock and CLKOUT. The EBDF field must not be changed during an external bus access or while an access is pending.<br><br>00 Divide by 1.<br>01 Divide by 2.<br>10 Divide by 3.<br>11 Divide by 4. |

**8.2.31 System Clock Register (SIU\_SYSDIV)**

The SIU\_SYSDIV register provides clock multiplexing control and clock frequency division factors to configure the system clocks. Please refer to the clocking section for a diagram showing how the fields of this register are used to control the various clock muxes and dividers.

**NOTE**

Do not change the contents of this register while the FEC is operating, otherwise transactions in the crossbar may be corrupted.

Address: 0h base + 9A0h offset = 9A0h

|       |         |    |            |    |         |    |            |    |          |    |        |    |    |    |    |    |
|-------|---------|----|------------|----|---------|----|------------|----|----------|----|--------|----|----|----|----|----|
| Bit   | 0       | 1  | 2          | 3  | 4       | 5  | 6          | 7  | 8        | 9  | 10     | 11 | 12 | 13 | 14 | 15 |
| R     | LCK     | 0  | PLLOSEL    | 0  | PLL1SEL | 0  | PERCLKSEL  | 0  | FMPERDIV | 0  | PERDIV |    |    |    |    |    |
| W     |         |    |            |    |         |    |            |    |          |    |        |    |    |    |    |    |
| Reset | 0       | 0  | 0          | 0  | 0       | 0  | 0          | 0  | 0        | 0  | 0      | 0  | 0  | 0  | 0  | 0  |
| Bit   | 16      | 17 | 18         | 19 | 20      | 21 | 22         | 23 | 24       | 25 | 26     | 27 | 28 | 29 | 30 | 31 |
| R     | MCANSEL | 0  | SYSCCLKSEL | 0  | ETPUDIV | 0  | SYSCCLKDIV | 0  | PCSEN    |    |        |    |    |    |    |    |
| W     |         |    |            |    |         |    |            |    |          |    |        |    |    |    |    |    |
| Reset | 0       | 0  | 0          | 0  | 0       | 0  | 0          | 0  | 0        | 0  | 0      | 1  | 0  | 0  | 0  | 0  |

## SIU\_SYSDIV field descriptions

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                      |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>LCK          | Write Lock Bit. When the LCK is written with a value of 1, writes to all other bits and fields in the SIU_SYSDIV register are disabled. The LCK bit can be set/cleared by software at any time.<br><br>0 SIU_SYSDIV register is unlocked and writes are permitted.<br>1 SIU_SYSDIV register is locked, and only writes to the LCK bit are permitted.                                             |
| 1–2<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                          |
| 3<br>PLL0SEL      | PLL0 Select. Selects the clock source for PLL0.<br><br>0 The crystal oscillator (XOSC) is the clock source for PLL0.<br>1 The internal RC oscillator (IRC) is the clock source for PLL0.                                                                                                                                                                                                         |
| 4<br>Reserved     | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                          |
| 5<br>PLL1SEL      | PLL1 Select. Selects the clock source for PLL1.<br><br>0 The crystal oscillator (XOSC) is the clock source for PLL1.<br>1 The PHI1 output of PLL0 is the clock source for PLL1.                                                                                                                                                                                                                  |
| 6<br>Reserved     | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                          |
| 7<br>PERCLKSEL    | Peripheral Clock Select. Selects the clock source for eTPU and other communication peripherals that require non-FM clock for their protocol engines.<br><br>0 Clock is connected to the FM clock domain (output of the SYSCLKDIV divider), for legacy support.<br>1 Clock is connected to the non-FM clock domain (output of PLL0).                                                              |
| 8–9<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                          |
| 10–11<br>FMPERDIV | FM Clock Peripheral Divider. Selects the divider value for the clock that feeds the memory mapped portion of the peripheral interfaces, and also feeds peripheral engines that can tolerate FM modulation.<br><br><b>CAUTION:</b> For proper device operation, always use the FMPERDIV setting 00b for divide by 2.<br><br>00 Divide by 2.<br>01 Divide by 4.<br>10 Divide by 8.<br>11 Reserved. |
| 12–13<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                          |
| 14–15<br>PERDIV   | Non-FM Clock Peripheral Divider. Selects the divider value for the clock that feeds protocol engines of peripherals that must be clocked by a stable non-FM clock.<br><br>00 Divide by 2.<br>01 Divide by 4.<br>10 Divide by 8.<br>11 Reserved.                                                                                                                                                  |
| 16<br>MCANSEL     | MCAN Select. Selects the clock source for both MCAN modules.<br><br>0 The crystal oscillator (XOSC) is the clock source for the MCAN modules.<br>1 The non-FM peripheral clock is the clock source for the MCAN modules.                                                                                                                                                                         |

Table continues on the next page...

**SIU\_SYSDIV field descriptions (continued)**

| Field              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 17<br>Reserved     | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 18–19<br>SYSCLKSEL | System Clock Select. The SYSCLKSEL field directly selects the source of the system clock if the Progressive Clock Switch (PCS) is disabled. If the PCS is enabled, then the SYSCLKSEL field is a request to change the clock, and the actual changing is controlled by the PCS. In this case, information about the actual clock source driving the system clock is available in the SIU_PCSIFR[PCSMS] field. Note that if the PCS fails to switch (because, for example, it was programmed incorrectly), then the system clock will not switch to the clock requested by SYSCLKSEL, and SIU_PCSIFR[PCSMS] will still indicate the previous clock.<br><br>00 The system clock is driven by the 16 MHz Internal RC Oscillator (IRC).<br>01 The system clock is driven by the external oscillator (XOSC).<br>10 The system clock is driven by the FM PLL (PLL1).<br>11 The system clock is driven by the non-FM PLL (PLL0). |
| 20–22<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 23<br>ETPUDIV      | eTPU Clock Divider. Selects the divider value for the eTPU clock.<br><br>0 Divide by 2.<br>1 Divide by 1.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 24–26<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 27–29<br>SYSCLKDIV | System Clock Divider. Selects the divider value for the system clock coming out of PLL1. Note that the SYSCLKDIV divider is required in addition to the PLL output divider to allow other sources for the system clock (IRC, OSC) to be divided down to slowest frequencies to enable very low power modes.<br><br>000 Divide by 2.<br>001 Divide by 4.<br>010 Divide by 8.<br>011 Divide by 16.<br>100 Divide by 1.<br>101 Divide by 1.<br>110 Divide by 1.<br>111 Divide by 1.                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 30<br>Reserved     | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 31<br>PCSEN        | Progressive Clock Switch Enable. Enables or disables the progressive clock switching (PCS) feature on the system clock.<br><br>0 PCS disabled.<br>1 PCS enabled.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

## 8.2.32 Halt Register 1 (SIU\_HLT1)

This register is used to dispatch a request to various modules to halt their clock inputs. Each register bit drives a separate halt request signal. Modules respond differently to the halt request. The CPUs will allow their clock inputs to be halted only after the execution of the WAIT instruction. Most other modules will wait for internal activities to finish before allowing their clock inputs to be halted, but a few modules will allow the clocks to be halted immediately.

Some peripherals have an MDIS (module disable) bit in the module control register that can be set to disable the module clock, reducing power consumption. In most cases the peripheral registers are still readable and writeable. However, using the SIU\_HALT1 register also disables the read/write functions of the disabled peripheral registers for additional power saving.

When the clocks of all CPUs have been halted and all cross-bar masters are idle, then the clocks of the following modules are also halted automatically: cross-bar, peripheral bridges, system RAM, DMA, STM and Semaphores. The interrupt controller and SWT\_A/SWT\_B clocks are never stopped.

Address: 0h base + 9A4h offset = 9A4h

| Bit   | 0     | 1     | 2 | 3   | 4     | 5     | 6   | 7 | 8      | 9      | 10     | 11 | 12     | 13  | 14 | 15 |
|-------|-------|-------|---|-----|-------|-------|-----|---|--------|--------|--------|----|--------|-----|----|----|
| R     | CORE0 | CORE1 | 0 | CSE | ETPUC | ETPUA | NPC | 0 | EQADCB | EQADCA | EMIOS0 | 0  | EMIOS1 | PIT | 0  |    |
| W     |       |       |   |     |       |       |     |   |        |        |        |    |        |     |    |    |
| Reset | 0     | 1     | 0 | 0   | 0     | 0     | 0   | 0 | 0      | 0      | 0      | 0  | 0      | 0   | 0  | 0  |

| Bit   | 16       | 17       | 18       | 19       | 20    | 21    | 22    | 23    | 24    | 25 | 26    | 27    | 28    | 29    | 30    | 31    |
|-------|----------|----------|----------|----------|-------|-------|-------|-------|-------|----|-------|-------|-------|-------|-------|-------|
| R     | FLEXCAND | FLEXCANC | FLEXCANB | FLEXCANA | DSPID | DSPIC | DSPIB | DSPIA | DSPIE | 0  | ESCIF | ESCIE | ESCID | ESCIC | ESCIB | ESCIA |
| W     |          |          |          |          |       |       |       |       |       |    |       |       |       |       |       |       |
| Reset | 0        | 0        | 0        | 0        | 0     | 0     | 0     | 0     | 0     | 0  | 0     | 0     | 0     | 0     | 0     | 0     |

**SIU\_HLT1 field descriptions**

| Field      | Description                                                                                |
|------------|--------------------------------------------------------------------------------------------|
| 0<br>CORE0 | Halt Core 0. Halts the clock inputs of Core 0 after the execution of the WAIT instruction. |
| 0          | No halt request.                                                                           |
| 1          | Dispatch halt request.                                                                     |

*Table continues on the next page...*



## SIU\_HLT1 field descriptions (continued)

| Field         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1<br>CORE1    | Halt Core 1. Halt the clock inputs of Core 1 after the execution of the WAIT instruction.<br><br>0 No halt request.<br>1 Dispatch halt request.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 2<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 3<br>CSE      | Halt CSE. Halt the clock inputs of the Cryptographic Services Engine. The clocks are halted immediately after this bit is set, therefore software must guarantee that the CSE module can be stopped prior to asserting the halt request.<br><br>0 No halt request.<br>1 Dispatch halt request.                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 4<br>ETPUC    | Halt eTPU C. Dispatch a halt request to Enhanced Timing Processing Unit C and to the Reaction Module. These modules will hold the request until all internal activities are concluded, then they will allow their clock inputs to be halted.<br><br>0 No halt request.<br>1 Dispatch halt request.                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 5<br>ETPUA    | Halt eTPU A/B. Dispatch a halt request to Enhanced Timing Processing Unit A/B. The module will hold the request until all internal activities are concluded, then it will allow its clock inputs to be halted.<br><br>0 No halt request.<br>1 Dispatch halt request.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 6<br>NPC      | Halt NPC. Dispatch a halt request to the Nexus Port Controller. The module will hold the request until all internal activities are concluded, then it will allow its clock inputs to be halted.<br><br><b>NOTE:</b> When this bit is 1 and the NPC clock is halted, if any of the following Nexus clients gets out of the IDLE state, the NPC clock is re-enabled temporarily: <ul style="list-style-type: none"> <li>• Core0</li> <li>• Core1</li> <li>• eDMA_A</li> <li>• eDMA_B</li> <li>• SIPI</li> </ul> <p>The NPC clock is re-enabled only during the period when any of these clients is out of the IDLE state. When all of these clients resume the IDLE state, the NPC clock is halted again automatically.</p><br>0 No halt request.<br>1 Dispatch halt request. |
| 7<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 8<br>EQADCB   | Halt eQADC B. Dispatch a halt request to the Enhanced Queued Analog-to-Digital Converter B. The module will hold the request until all internal activities are concluded, then it will allow its clock inputs to be halted. This halt request prompts eQADC_B to enter the module's Stop mode.<br><br>0 No halt request.<br>1 Dispatch halt request.                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 9<br>EQADCA   | Halt eQADC A. Dispatch a halt request to the Enhanced Queued Analog-to-Digital Converter A. The module will hold the request until all internal activities are concluded, then it will allow its clock inputs to be halted. This halt request prompts eQADC_A to enter the module's Stop mode.                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

Table continues on the next page...

**SIU\_HLT1 field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                            |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | 0 No halt request.<br>1 Dispatch halt request.                                                                                                                                                                                                                                                         |
| 10<br>EMIOS0      | Halt eMIOS 0. Halt the clock inputs of the Enhanced Modular I/O System 0. The clocks are halted immediately after this bit is set, therefore software must guarantee that the eMIOS_0 module can be stopped prior to asserting the halt request.<br><br>0 No halt request.<br>1 Dispatch halt request. |
| 11<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                |
| 12<br>EMIOS1      | Halt eMIOS 1. Halt the clock inputs of the Enhanced Modular I/O System 1. The clocks are halted immediately after this bit is set, therefore software must guarantee that the eMIOS_1 module can be stopped prior to asserting the halt request.<br><br>0 No halt request.<br>1 Dispatch halt request. |
| 13<br>PIT         | Halt PIT. Dispatch a halt request to the Periodic Interval Timer. The module will hold the request until all internal activities are concluded, then it will allow its clock inputs to be halted.<br><br>0 No halt request.<br>1 Dispatch halt request.                                                |
| 14–15<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                |
| 16<br>FLEXCAN D   | Halt FlexCAN D. Dispatch a halt request to the Flexible Controller Area Network D. The module will hold the request until all internal activities are concluded, then it will allow its clock inputs to be halted.<br><br>0 No halt request.<br>1 Dispatch halt request.                               |
| 17<br>FLEXCAN C   | Halt FlexCAN C. Dispatch a halt request to the Flexible Controller Area Network C. The module will hold the request until all internal activities are concluded, then it will allow its clock inputs to be halted.<br><br>0 No halt request.<br>1 Dispatch halt request.                               |
| 18<br>FLEXCAN B   | Halt FlexCAN B. Dispatch a halt request to the Flexible Controller Area Network B. The module will hold the request until all internal activities are concluded, then it will allow its clock inputs to be halted.<br><br>0 No halt request.<br>1 Dispatch halt request.                               |
| 19<br>FLEXCAN A   | Halt FlexCAN A. Dispatch a halt request to the Flexible Controller Area Network A. The module will hold the request until all internal activities are concluded, then it will allow its clock inputs to be halted.<br><br>0 No halt request.<br>1 Dispatch halt request.                               |
| 20<br>DSPID       | Halt DSPI D. Dispatch a halt request to the Deserial/Serial Peripheral Interface D. The module will hold the request until all internal activities are concluded, then it will allow its clock inputs to be halted.<br><br>0 No halt request.<br>1 Dispatch halt request.                              |
| 21<br>DSPIC       | Halt DSPI C. Dispatch a halt request to the Deserial/Serial Peripheral Interface C. The module will hold the request until all internal activities are concluded, then it will allow its clock inputs to be halted.                                                                                    |

*Table continues on the next page...*

**SIU\_HLT1 field descriptions (continued)**

| <b>Field</b>   | <b>Description</b>                                                                                                                                                                                                                                                           |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | 0 No halt request.<br>1 Dispatch halt request.                                                                                                                                                                                                                               |
| 22<br>DSPIB    | Halt DSPI B. Dispatch a halt request to the Deserial/Serial Peripheral Interface B. The module will hold the request until all internal activities are concluded, then it will allow its clock inputs to be halted.<br><br>0 No halt request.<br>1 Dispatch halt request.    |
| 23<br>DSPIA    | Halt DSPI A. Dispatch a halt request to the Deserial/Serial Peripheral Interface A. The module will hold the request until all internal activities are concluded, then it will allow its clock inputs to be halted.<br><br>0 No halt request.<br>1 Dispatch halt request.    |
| 24<br>DSPIE    | Halt DSPI E. Dispatch a halt request to the Deserial/Serial Peripheral Interface E. The module will hold the request until all internal activities are concluded, then it will allow its clock inputs to be halted.<br><br>0 No halt request.<br>1 Dispatch halt request.    |
| 25<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                      |
| 26<br>ESCIF    | Halt eSCI F. Dispatch a halt request to the Enhanced Serial Communication Interface F. The module will hold the request until all internal activities are concluded, then it will allow its clock inputs to be halted.<br><br>0 No halt request.<br>1 Dispatch halt request. |
| 27<br>ESCIE    | Halt eSCI E. Dispatch a halt request to the Enhanced Serial Communication Interface E. The module will hold the request until all internal activities are concluded, then it will allow its clock inputs to be halted.<br><br>0 No halt request.<br>1 Dispatch halt request. |
| 28<br>ESCID    | Halt eSCI D. Dispatch a halt request to the Enhanced Serial Communication Interface D. The module will hold the request until all internal activities are concluded, then it will allow its clock inputs to be halted.<br><br>0 No halt request.<br>1 Dispatch halt request. |
| 29<br>ESCIC    | Halt eSCI C. Dispatch a halt request to the Enhanced Serial Communication Interface C. The module will hold the request until all internal activities are concluded, then it will allow its clock inputs to be halted.<br><br>0 No halt request.<br>1 Dispatch halt request. |
| 30<br>ESCIB    | Halt eSCI B. Dispatch a halt request to the Enhanced Serial Communication Interface B. The module will hold the request until all internal activities are concluded, then it will allow its clock inputs to be halted.<br><br>0 No halt request.<br>1 Dispatch halt request. |
| 31<br>ESCIA    | Halt eSCI A. Dispatch a halt request to the Enhanced Serial Communication Interface A. The module will hold the request until all internal activities are concluded, then it will allow its clock inputs to be halted.<br><br>0 No halt request.<br>1 Dispatch halt request. |

8.2.33 Halt Acknowledge Register 1 (SIU\_HLTACK1)

The SIU\_HLTACK1 bits indicate that the request to halt the modules via the respective SIU\_HLT1 bits have completed and the clock inputs to the modules have been disabled.

Address: 0h base + 9A8h offset = 9A8h

|       |          |          |          |          |       |       |       |       |        |        |        |       |        |       |       |       |
|-------|----------|----------|----------|----------|-------|-------|-------|-------|--------|--------|--------|-------|--------|-------|-------|-------|
| Bit   | 0        | 1        | 2        | 3        | 4     | 5     | 6     | 7     | 8      | 9      | 10     | 11    | 12     | 13    | 14    | 15    |
| R     | CORE0    | CORE1    | 0        | CSE      | ETPUC | ETPUA | NPC   | 0     | EQADCB | EQADCA | EMIOS0 | 0     | EMIOS1 | PIT   | 0     |       |
| W     |          |          |          |          |       |       |       |       |        |        |        |       |        |       |       |       |
| Reset | 0        | 0        | 0        | 0        | 0     | 0     | 0     | 0     | 0      | 0      | 0      | 0     | 0      | 0     | 0     | 0     |
| Bit   | 16       | 17       | 18       | 19       | 20    | 21    | 22    | 23    | 24     | 25     | 26     | 27    | 28     | 29    | 30    | 31    |
| R     | FLEXCAND | FLEXCANC | FLEXCANB | FLEXCANA | DSPID | DSPIC | DSPIB | DSPIA | DSPIE  | 0      | ESCIF  | ESCIE | ESCID  | ESCIC | ESCIB | ESCIA |
| W     |          |          |          |          |       |       |       |       |        |        |        |       |        |       |       |       |
| Reset | 0        | 0        | 0        | 0        | 0     | 0     | 0     | 0     | 0      | 0      | 0      | 0     | 0      | 0     | 0     | 0     |

SIU\_HLTACK1 field descriptions

| Field      | Description              |
|------------|--------------------------|
| 0<br>CORE0 | Halt Acknowledge Core 0. |

Table continues on the next page...

**SIU\_HLTACK1 field descriptions (continued)**

| Field          | Description                                                                                                         |
|----------------|---------------------------------------------------------------------------------------------------------------------|
|                | 0 Not halted.<br>1 Halted.                                                                                          |
| 1<br>CORE1     | Halt Acknowledge Core 1.<br><br>0 Not halted.<br>1 Halted.                                                          |
| 2<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                             |
| 3<br>CSE       | Halt Acknowledge CSE.<br><br>0 Not halted.<br>1 Halted.                                                             |
| 4<br>ETPUC     | Halt Acknowledge eTPU C and Reaction Module.<br><br>0 Not halted.<br>1 Both eTPU C and Reaction Modules are Halted. |
| 5<br>ETPUA     | Halt Acknowledge eTPU A/B.<br><br>0 Not halted.<br>1 Halted.                                                        |
| 6<br>NPC       | Halt Acknowledge NPC.<br><br>0 Not halted.<br>1 Halted.                                                             |
| 7<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                             |
| 8<br>EQADCB    | Halt Acknowledge eQADC B.<br><br>0 Not halted.<br>1 Halted.                                                         |
| 9<br>EQADCA    | Halt Acknowledge eQADC A.<br><br>0 Not halted.<br>1 Halted.                                                         |
| 10<br>EMIOS0   | Halt Acknowledge eMIOS 0.<br><br>0 Not halted.<br>1 Halted.                                                         |
| 11<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                             |
| 12<br>EMIOS1   | Halt Acknowledge eMIOS 1.<br><br>0 Not halted.<br>1 Halted.                                                         |
| 13<br>PIT      | Halt Acknowledge PIT.<br><br>0 Not halted.<br>1 Halted.                                                             |

*Table continues on the next page...*

**SIU\_HLTACK1 field descriptions (continued)**

| Field             | Description                                                                             |
|-------------------|-----------------------------------------------------------------------------------------|
| 14–15<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 16<br>FLEXCAND    | Halt Acknowledge FlexCAN D.<br>0 Not halted.<br>1 Halted.                               |
| 17<br>FLEXCANC    | Halt Acknowledge FlexCAN C.<br>0 Not halted.<br>1 Halted.                               |
| 18<br>FLEXCANB    | Halt Acknowledge FlexCAN B.<br>0 Not halted.<br>1 Halted.                               |
| 19<br>FLEXCANA    | Halt Acknowledge FlexCAN A.<br>0 Not halted.<br>1 Halted.                               |
| 20<br>DSPID       | Halt Acknowledge DSPI D.<br>0 Not halted.<br>1 Halted.                                  |
| 21<br>DSPIC       | Halt Acknowledge DSPI C.<br>0 Not halted.<br>1 Halted.                                  |
| 22<br>DSPIB       | Halt Acknowledge DSPI B.<br>0 Not halted.<br>1 Halted.                                  |
| 23<br>DSPIA       | Halt Acknowledge DSPI A.<br>0 Not halted.<br>1 Halted.                                  |
| 24<br>DSPIE       | Halt Acknowledge DSPI E.<br>0 Not halted.<br>1 Halted.                                  |
| 25<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 26<br>ESCIF       | Halt Acknowledge eSCI F.<br>0 Not halted.<br>1 Halted.                                  |
| 27<br>ESCIE       | Halt Acknowledge eSCI E.<br>0 Not halted.<br>1 Halted.                                  |

*Table continues on the next page...*

**SIU\_HLTACK1 field descriptions (continued)**

| Field       | Description                                            |
|-------------|--------------------------------------------------------|
| 28<br>ESCID | Halt Acknowledge eSCI D.<br>0 Not halted.<br>1 Halted. |
| 29<br>ESCIC | Halt Acknowledge eSCI C.<br>0 Not halted.<br>1 Halted. |
| 30<br>ESCIB | Halt Acknowledge eSCI B.<br>0 Not halted.<br>1 Halted. |
| 31<br>ESCIA | Halt Acknowledge eSCI A.<br>0 Not halted.<br>1 Halted. |

**8.2.34 Core0 Reset Vector Register (SIU\_RSTVEC0)**

This register contains the reset vector, reset control and instruction execution type selection for Core0. The RST bit is qualified by the CORE0 halt bit in the SIU\_HLT1 register, such that both must be 1 for a core reset to occur. The order in which the RST and core halt bits are asserted does not matter. After both halt and RST bits are asserted, internal logic places a request to halt the corresponding core clocks, but the core may continue to run until such time that it terminates any pending crossbar master bus cycles and interrupt controller interaction that might prevent their use by other bus masters. When all pending transactions finish, the core will be in reset state with clock stopped. Exit of reset state is done by negating either SIU\_HLT1[CORE0] or the RST bit. Negating just one is enough to remove the core from reset state.

The reset value of the halt and RST bits for Core0 is 0 by default to allow Core0 to execute out of reset. The halt and RST bits for Core1 are 1 by default to prevent Core1 from executing out of reset.

Any attempt to reset both cores by setting both RST bits will result in an immediate system reset, with the cause of the reset recorded in the SIU\_RSR[CPURS] bit.

**NOTE**

The halt acknowledge bit in the SIU\_HLTACK1 register will not be asserted for a core in reset state. Even though setting the halt bit is part of the procedure to put a core in reset state, the halt acknowledge bit is only asserted when the WAIT

instruction is executed and the core is stopped, but not in reset state.

Address: 0h base + 9ACh offset = 9ACh

|       |        |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|--------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0      | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | RSTVEC |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |        |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 1      | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1  | 1  | 1  | 1  | 1  | 1  |

|       |        |    |    |    |    |    |    |    |    |    |    |    |    |    |     |     |
|-------|--------|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|-----|
| Bit   | 16     | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30  | 31  |
| R     | RSTVEC |    |    |    |    |    |    |    |    |    |    |    |    |    | RST | VLE |
| W     |        |    |    |    |    |    |    |    |    |    |    |    |    |    |     |     |
| Reset | 1      | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 0   | 0   |

### SIU\_RSTVEC0 field descriptions

| Field          | Description                                                                                                                                                                                                                                                                                                                                             |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–29<br>RSTVEC | Reset Vector. The RSTVEC value determines the initial program counter upon exiting reset. The MMU TLB entry 0 effective and real page number (EPN/RPN) is automatically loaded with RSTVEC[0:21] when reset is released and the TLB entry size is set to 4KB. In the e200z759 Core Reference Manual the reset vector is referred to as p_rstbase[0-21]. |
| 30<br>RST      | Controls the assertion of RESET to the core. Writing 1 to this bit causes the core to enter reset, provided the respective SIU_HLT1 bit is also set. Reads of this bit indicate whether the core is being held in reset.<br><br>0 Core is not in reset.<br>1 Reset request.                                                                             |
| 31<br>VLE      | VLE Select. The VLE bit selects whether the core executes VLE or Book E code at the reset vector address.<br><br>0 Book E.<br>1 VLE.                                                                                                                                                                                                                    |

## 8.2.35 Core1 Reset Vector Register (SIU\_RSTVEC1)

This register contains the reset vector, reset control and instruction execution type selection for Core1. The RST bit is qualified by the CORE1 halt bit in the SIU\_HLT1 register, such that both must be 1 for a core reset to occur. The order in which the RST and core halt bits are asserted does not matter. After both halt and RST bits are asserted, internal logic places a request to halt the corresponding core clocks, but the core may continue to run until such time that it terminates any pending crossbar master bus cycles and interrupt controller interaction that might prevent their use by other bus masters. When all pending transactions finish, the core will be in reset state with clock stopped. Exit of reset state is done by negating either SIU\_HLT1[CORE1] or the RST bit. Negating just one is enough to remove the core from reset state.

The reset value of the halt and RST bits for Core0 is 0 by default to allow Core0 to execute out of reset. The halt and RST bits for Core1 are 1 by default to prevent Core1 from executing out of reset.



Any attempt to reset both cores by setting both RST bits will result in an immediate system reset, with the cause of the reset recorded in the SIU\_RSR[CPURS] bit.

### NOTE

The halt acknowledge bit in the SIU\_HLTACK1 register will not be asserted for a core in reset state. Even though setting the halt bit is part of the procedure to put a core in reset state, the halt acknowledge bit is only asserted when the WAIT instruction is executed and the core is stopped, but not in reset state.

Address: 0h base + 9B0h offset = 9B0h

|       |        |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|--------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0      | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | RSTVEC |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |        |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 1      | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1  | 1  | 1  | 1  | 1  | 1  |

|       |        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |     |
|-------|--------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|-----|
| Bit   | 16     | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31  |     |
| R     | RSTVEC |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RST | VLE |
| W     |        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |     |
| Reset | 1      | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 0   |     |

### SIU\_RSTVEC1 field descriptions

| Field          | Description                                                                                                                                                                                                                                                                                                                                             |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–29<br>RSTVEC | Reset Vector. The RSTVEC value determines the initial program counter upon exiting reset. The MMU TLB entry 0 effective and real page number (EPN/RPN) is automatically loaded with RSTVEC[0:21] when reset is released and the TLB entry size is set to 4KB. In the e200z759 Core Reference Manual the reset vector is referred to as p_rstbase[0-21]. |
| 30<br>RST      | Controls the assertion of RESET to the core. Writing 1 to this bit causes the core to enter reset, provided the respective SIU_HLT1 bit is also set. Reads of this bit indicate whether the core is being held in reset.<br><br>0 Core is not in reset.<br>1 Reset request.                                                                             |
| 31<br>VLE      | VLE Select. The VLE bit selects whether the core executes VLE or Book E code at the reset vector address.<br><br>0 Book E.<br>1 VLE.                                                                                                                                                                                                                    |

## 8.2.36 Core0 PID mapping control register (SIU\_C0PID)

This register allows an external tool to modify the logical to physical address mapping without interrupting normal application code execution in either core. An optional synchronization mechanism using the nex\_wevto[2] watchpoint event output from each core allows the mapping that is selected by the tool to change deterministically, when a specified instruction address is reached or a specified load/store address is accessed. If the synchronization mechanism is not enabled, a write to the register is propagated immediately to the MMU.

Address: 0h base + 9B4h offset = 9B4h

|       |            |   |              |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|------------|---|--------------|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0          | 1 | 2            | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | EXT_PID_EN |   | EXT_PID_SYNC |   | 0 |   |   |   |   |   |    |    |    |    |    |    |
| W     |            |   |              |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0          | 0 | 0            | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |         |  |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--|--|--|--|--|--|--|--|--|---------|--|
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |  |  |  |  |  |  |  |  |  |         |  |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  | EXT_PID |  |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |         |  |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |  |  |  |  |  |  |  |  |  |         |  |

### SIU\_C0PID field descriptions

| Field             | Description                                                                                                                                                                                                                                                                                                  |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>EXT_PID_EN   | External PID Selection Enable. Enable the values of the EXT_PID field to be routed to the MMU.<br>0 The contents of this register are not used to select the alternate MMU mapping.<br>1 The contents of the EXT_PID field are used to select the alternate MMU mapping.                                     |
| 1<br>EXT_PID_SYNC | External PID Synchronization. This bit configures whether the Nexus Watchpoint Event 2 is used or not to transfer the EXT_PID values to the MMU.<br>0 The Nexus Watchpoint Event 2 does not transfer the EXT_PID field to the MMU.<br>1 The Nexus Watchpoint Event 2 transfers the EXT_PID field to the MMU. |
| 2–29<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                      |
| 30–31<br>EXT_PID  | External PID. These are the external PID bits that are routed by the MMU to implement an alternate logical to physical address mapping under control of an external tool.                                                                                                                                    |

### 8.2.37 Core1 PID mapping control register (SIU\_C1PID)

This register allows an external tool to modify the logical to physical address mapping without interrupting normal application code execution in either core. An optional synchronization mechanism using the nex\_wevto[2] watchpoint event output from each core allows the mapping that is selected by the tool to change deterministically, when a specified instruction address is reached or a specified load/store address is accessed. If the synchronization mechanism is not enabled, a write to the register is propagated immediately to the MMU.

Address: 0h base + 9B8h offset = 9B8h

|       |            |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |         |  |
|-------|------------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--|--|--|--|--|--|--|--|--|--|--|--|--|---------|--|
| Bit   | 0          | 1            | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |  |  |  |  |  |  |  |  |  |  |  |  |  |         |  |
| R     | EXT_PID_EN | EXT_PID_SYNC | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |         |  |
| W     |            |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |         |  |
| Reset | 0          | 0            | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |  |  |  |  |  |  |  |  |  |  |  |  |  |         |  |
| Bit   | 16         | 17           | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |  |  |  |  |  |  |  |  |  |  |  |  |  |         |  |
| R     | 0          |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  | EXT_PID |  |
| W     |            |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |         |  |
| Reset | 0          | 0            | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |  |  |  |  |  |  |  |  |  |  |  |  |  |         |  |

#### SIU\_C1PID field descriptions

| Field             | Description                                                                                                                                                                                                                                                                                                  |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>EXT_PID_EN   | External PID Selection Enable. Enable the values of the EXT_PID field to be routed to the MMU.<br>0 The contents of this register are not used to select the alternate MMU mapping.<br>1 The contents of the EXT_PID field are used to select the alternate MMU mapping.                                     |
| 1<br>EXT_PID_SYNC | External PID Synchronization. This bit configures whether the Nexus Watchpoint Event 2 is used or not to transfer the EXT_PID values to the MMU.<br>0 The Nexus Watchpoint Event 2 does not transfer the EXT_PID field to the MMU.<br>1 The Nexus Watchpoint Event 2 transfers the EXT_PID field to the MMU. |
| 2–29<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                      |
| 30–31<br>EXT_PID  | External PID. These are the external PID bits that are routed by the MMU to implement an alternate logical to physical address mapping under control of an external tool.                                                                                                                                    |

## 8.2.38 Halt Register 2 (SIU\_HLT2)

This register is used to dispatch a request to various modules to halt their clock inputs. Each register bit drives a separate halt request signal. Modules respond differently to the halt request. Most modules will wait for internal activities to finish before allowing their clock inputs to be halted, but a few modules will allow the clocks to be halted immediately.

Some peripherals have an MDIS (module disable) bit in the module control register that can be set to disable the module clock, reducing power consumption. In most cases the peripheral registers are still readable and writeable. However, using the SIU\_HALT2 register also disables the read/write functions of the disabled peripheral registers for additional power saving.

When the clocks of all CPUs have been halted and all cross-bar masters are idle, then the clocks of the following modules are also halted automatically: cross-bar, peripheral bridges, system RAM, DMA, STM and Semaphores. The interrupt controller and SWT\_A/SWT\_B clocks are never stopped.

Address: 0h base + 9BCh offset = 9BCh

|       |     |     |      |    |    |    |      |      |    |    |    |    |     |     |     |       |
|-------|-----|-----|------|----|----|----|------|------|----|----|----|----|-----|-----|-----|-------|
| Bit   | 0   | 1   | 2    | 3  | 4  | 5  | 6    | 7    | 8  | 9  | 10 | 11 | 12  | 13  | 14  | 15    |
| R     | 0   |     |      |    |    |    |      |      |    |    |    |    | SDD | SDC | SDB | SDA   |
| W     | FEC |     |      |    |    |    |      |      |    |    |    |    |     |     |     |       |
| Reset | 0   | 0   | 0    | 0  | 0  | 0  | 0    | 0    | 0  | 0  | 0  | 0  | 0   | 0   | 0   | 0     |
| Bit   | 16  | 17  | 18   | 19 | 20 | 21 | 22   | 23   | 24 | 25 | 26 | 27 | 28  | 29  | 30  | 31    |
| R     | 0   |     | STCU | 0  |    |    | SRX1 | SRX0 | 0  | 0  | 0  | 0  | 0   |     |     | MCANB |
| W     |     | CRC |      |    |    |    |      |      |    |    |    |    |     |     |     | MCANA |
| Reset | 0   | 0   | 0    | 0  | 0  | 0  | 0    | 0    | 0  | 0  | 0  | 0  | 0   | 0   | 0   | 0     |

**SIU\_HLT2 field descriptions**

| Field            | Description                                                                                                                                                                                                                                                                               |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>FEC         | Halt FEC. Halt the clock inputs of the Fast Ethernet Controller. The clocks are halted immediately after this bit is set, therefore software must guarantee that the FEC module can be stopped prior to asserting the halt request.<br><br>0 No halt request.<br>1 Dispatch halt request. |
| 1–11<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                   |

*Table continues on the next page...*

**SIU\_HLT2 field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                              |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 12<br>SDD         | Halt Sigma Delta D. Dispatch a halt request to Sigma Delta D. The module will hold the request until all internal activities are concluded, then it will allow its clock inputs to be halted (digital and analog).<br><br>0 No halt request.<br>1 Dispatch halt request. |
| 13<br>SDC         | Halt Sigma Delta C. Dispatch a halt request to Sigma Delta C. The module will hold the request until all internal activities are concluded, then it will allow its clock inputs to be halted (digital and analog).<br><br>0 No halt request.<br>1 Dispatch halt request. |
| 14<br>SDB         | Halt Sigma Delta B. Dispatch a halt request to Sigma Delta B. The module will hold the request until all internal activities are concluded, then it will allow its clock inputs to be halted (digital and analog).<br><br>0 No halt request.<br>1 Dispatch halt request. |
| 15<br>SDA         | Halt Sigma Delta A. Dispatch a halt request to Sigma Delta A. The module will hold the request until all internal activities are concluded, then it will allow its clock inputs to be halted (digital and analog).<br><br>0 No halt request.<br>1 Dispatch halt request. |
| 16<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                  |
| 17<br>CRC         | Halt CRC. Halt the clocks to the CRC. The clocks are halted immediately after this bit is set, therefore software must guarantee that the CRC can be stopped prior to asserting the halt request.<br><br>0 No halt request.<br>1 Dispatch halt request.                  |
| 18<br>STCU        | Halt STCU. Halt the clocks to the STCU. The clocks are halted immediately after this bit is set, therefore software must guarantee that the STCU can be stopped prior to asserting the halt request.<br><br>0 No halt request.<br>1 Dispatch halt request.               |
| 19–21<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                  |
| 22<br>SRX1        | Halt SENT 1. Dispatch a halt request to the SENT 1. The module will hold the request until all internal activities are concluded, then it will allow its clock inputs to be halted.<br><br>0 No halt request.<br>1 Dispatch halt request.                                |
| 23<br>SRX0        | Halt SENT 0. Dispatch a halt request to the SENT 0. The module will hold the request until all internal activities are concluded, then it will allow its clock inputs to be halted.<br><br>0 No halt request.<br>1 Dispatch halt request.                                |
| 24–25<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                  |
| 26<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                  |
| 27<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                  |

*Table continues on the next page...*

**SIU\_HLT2 field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                           |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 28–29<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                               |
| 30<br>MCANB       | Halt MCAN B. Dispatch a halt request to MCAN B. The module will hold the request until all internal activities are concluded, then it will allow its clock inputs to be halted.<br><br>0 No halt request.<br>1 Dispatch halt request. |
| 31<br>MCANA       | Halt MCAN A. Dispatch a halt request to MCAN A. The module will hold the request until all internal activities are concluded, then it will allow its clock inputs to be halted.<br><br>0 No halt request.<br>1 Dispatch halt request. |

### 8.2.39 Halt Acknowledge Register 2 (SIU\_HLTACK2)

The SIU\_HLTACK2 bits indicate that the request to halt the modules via the respective SIU\_HLT2 bits have completed and the clock inputs to the modules have been disabled.

Address: 0h base + 9C0h offset = 9C0h

| Bit   | 0   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12  | 13  | 14  | 15  |
|-------|-----|---|---|---|---|---|---|---|---|---|----|----|-----|-----|-----|-----|
| R     | FEC |   |   |   |   |   | 0 |   |   |   |    |    | SDD | SDC | SDB | SDA |
| W     |     |   |   |   |   |   |   |   |   |   |    |    |     |     |     |     |
| Reset | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0   | 0   | 0   | 0   |

| Bit   | 16 | 17  | 18   | 19 | 20 | 21 | 22   | 23   | 24 | 25 | 26 | 27 | 28 | 29 | 30    | 31    |
|-------|----|-----|------|----|----|----|------|------|----|----|----|----|----|----|-------|-------|
| R     | 0  | CRC | STCU |    | 0  |    | SRX1 | SRX0 | 0  | 0  | 0  |    | 0  |    | MCANB | MCANA |
| W     |    |     |      |    |    |    |      |      |    |    |    |    |    |    |       |       |
| Reset | 0  | 0   | 0    | 0  | 0  | 0  | 0    | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0     | 0     |

SIU\_HLTACK2 field descriptions

| Field    | Description                                         |
|----------|-----------------------------------------------------|
| 0<br>FEC | Halt Acknowledge FEC.<br>0 Not halted.<br>1 Halted. |

Table continues on the next page...

## SIU\_HLTACK2 field descriptions (continued)

| Field             | Description                                                                             |
|-------------------|-----------------------------------------------------------------------------------------|
| 1–11<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 12<br>SDD         | Halt Acknowledge Sigma Delta D.<br><br>0 Not halted.<br>1 Halted.                       |
| 13<br>SDC         | Halt Acknowledge Sigma Delta C.<br><br>0 Not halted.<br>1 Halted.                       |
| 14<br>SDB         | Halt Acknowledge Sigma Delta B.<br><br>0 Not halted.<br>1 Halted.                       |
| 15<br>SDA         | Halt Acknowledge Sigma Delta A.<br><br>0 Not halted.<br>1 Halted.                       |
| 16<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 17<br>CRC         | Halt Acknowledge CRC.<br><br>0 Not halted.<br>1 Halted.                                 |
| 18<br>STCU        | Halt Acknowledge STCU.<br><br>0 Not halted.<br>1 Halted.                                |
| 19–21<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 22<br>SRX1        | Halt Acknowledge SENT 1.<br><br>0 Not halted.<br>1 Halted.                              |
| 23<br>SRX0        | Halt Acknowledge SENT 0.<br><br>0 Not halted.<br>1 Halted.                              |
| 24–25<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 26<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 27<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 28–29<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 30<br>MCANB       | Halt Acknowledge MCAN B.                                                                |

*Table continues on the next page...*



**SIU\_HLTACK2 field descriptions (continued)**

| Field       | Description                                                |
|-------------|------------------------------------------------------------|
|             | 0 Not halted.<br>1 Halted.                                 |
| 31<br>MCANA | Halt Acknowledge MCAN A.<br><br>0 Not halted.<br>1 Halted. |

**8.2.40 Sigma-Delta Clock Configuration Register (SIU\_SDCLKCFG)**

The SIU\_SDCLKCFG register defines the clock division factor to generate the clock to the Sigma-Delta ADCs. shows the SDDIV field in the clock generation scheme.

Address: 0h base + 9C4h offset = 9C4h

|       |     |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|-----|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | LCK |   |   |   |   |   |   |   | 0 |   |    |    |    |    |    |    |
| W     |     |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |    |    |    |    | 0  |    |    |    |    |    |    |    |    |    |    |    |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**SIU\_SDCLKCFG field descriptions**

| Field            | Description                                                                                                                                                                                                                                                                                                                                                |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>LCK         | Write Lock Bit. When the LCK is written with a value of 1, writes to all other bits and fields in the SIU_SDCLKCFG register are disabled. The LCK bit can be set/cleared by software at any time.<br><br>0 SIU_SDCLKCFG register is unlocked and writes are permitted.<br>1 SIU_SDCLKCFG register is locked, and only writes to the LCK bit are permitted. |
| 1–24<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                    |
| 25–31<br>SDDIV   | Sigma Delta Clock Divider. Selects the divider value for the clock that feeds the Sigma Delta ADCs, according to the following formula:<br><br>DivisionFactor = SDDIV + 1                                                                                                                                                                                  |

## 8.2.41 Reset Configuration Register (SIU\_RCR)

SIU\_RCR contains a field that configures reset operation.

Address: 0h base + 9D0h offset = 9D0h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27  | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RET |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0* | 0* | 0* | 0* |

\* Notes:

- RET field: RET is loaded with 1000b when the REE bit of the UTEST Reset Escalation Configuration DCF client is 1.

### SIU\_RCR field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–27<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                      |
| 28–31<br>RET     | Reset Escalation Threshold. This field defines the number of reset cycles not caused by POR that the device can execute before being held permanently in reset state, waiting for a POR. The defined number of non-POR reset cycles is the decimal value of RET plus 1, except when RET is 0. A value of 0 in this field disables the reset escalation feature. Any CPU write to the RET field re-initiates the escalation counting process. |

## 8.2.42 Lock-Step Status Register (SIU\_LOCKSTEP)

The SIU\_LOCKSTEP register contains a status bit that indicates whether the checker core is enabled or not. The checker core is enabled by setting the LOCKSTEP\_EN in the Miscellaneous DCF client. A reset cycle is required for a change in LOCKSTEP\_EN to take effect.

Address: 0h base + 9D4h offset = 9D4h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31  |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    | LSE |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   |

**SIU\_LOCKSTEP field descriptions**

| Field            | Description                                                                                                                                            |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–30<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                |
| 31<br>LSE        | Lock Step Enable. Indicates whether the checker core is enabled or not.<br><br>0 Lock step operation is disabled.<br>1 Lock step operation is enabled. |

**8.2.43 Progressive Clock Switch Interrupt Enable Register (SIU\_PCSE)**

The SIU\_PCSE controls the generation of an interrupt request when the SIU\_PCSIFR[PCSI] becomes set at the end of a progressive clock switch (PCS) operation.

Address: 0h base + 9D8h offset = 9D8h

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |      |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--|--|--|--|--|--|--|--|--|--|--|--|--|--|------|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |      |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |      |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |      |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |      |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |      |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  | PCSE |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |      |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |      |

**SIU\_PCSE field descriptions**

| Field            | Description                                                                                                                                                                                                                     |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–30<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                         |
| 31<br>PCSE       | PCS Interrupt Enable. Enables or disables the generation of an interrupt request when the SIU_PCSIFR[PCSI] becomes set at the end of a progressive clock switch operation.<br><br>0 Interrupt disabled.<br>1 Interrupt enabled. |

## 8.2.44 Progressive Clock Switch Interrupt Flag Register (SIU\_PCSIFR)

The SIU\_PCSIFR contains an interrupt flag that is set when the progressive clock switch operation finishes.

Address: 0h base + 9DCh offset = 9DCh

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |    |    |    |    |    |    |    |    |    |    |    |       |    |      |    |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|-------|----|------|----|
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28    | 29 | 30   | 31 |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    | PCSMS |    | PCSI |    |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |       |    | w1c  |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0     | 0  | 0    | 0  |

### SIU\_PCSIFR field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–28<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 29–30<br>PCSMS   | PCS Status. Indicates the clock source which effectively drives the system clock. Note that the clock source indicated by PCSMS may be different than the one requested by SIU_SYSDIV[SYSCLKSEL] because the PCS may still be active, or because the PCS may have failed to switch to the new clock (for example, if it was programmed incorrectly). After the PCSI flag gets set, indicating that the PCS operation is finished, software can read the value of the PCSMS field to find out if the PCS switch was successful.<br><br>00 The system clock is driven by the 16 MHz Internal RC Oscillator (IRC).<br>01 The system clock is driven by the external oscillator (XOSC).<br>10 The system clock is driven by the FM PLL (PLL1).<br>11 The system clock is driven by the non-FM PLL (PLL0). |
| 31<br>PCSI       | PCS Interrupt Flag. Indicates the end of a progressive clock switch operation. Must be cleared by writing '1' to the bit.<br><br>0 No PCS operation finished since last time this flag was cleared.<br>1 PCS operation finished.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |

### 8.2.45 Fast Ethernet Controller Control Register (SIU\_FECCR)

The SIU\_FECCR is used to select between the Media Independent Interface Lite (MII-Lite) and the Reduced Media Independent Interface (RMII) modes of the Fast Ethernet Controller (FEC).

Address: 0h base + 9E0h offset = 9E0h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | FM |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**SIU\_FECCR field descriptions**

| Field            | Description                                                                                                                                                                                                                                                                               |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–30<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                   |
| 31<br>FM         | FEC Mode. Selects between MII-Lite and RMII modes in the FEC module. Note that this bit does not affect the configuration of pads associated with the FEC operation. Pad configuration must be done in the PCR registers.<br><br>0 RMII mode is selected.<br>1 MII-Lite mode is selected. |

### 8.2.46 ECC Error Injection Register (SIU\_ECCEIR)

The SIU\_ECCEIR is used to inject ECC errors into reading operations on the MCAN RAM array. This register is intended to support testing of the error reporting path from the MCAN ECC logic to the FCCU input. Before programming this register, it is recommended to disable MCAN operations and then access the MCAN memories directly with the CPU.

ECC errors will be introduced into the read data path, therefore the memory contents remain intact. As soon as the error injection is disabled, normal MCAN operation can be resumed without the need to fix any memory content.

## Memory Map and Register Definition

Address: 0h base + 9E4h offset = 9E4h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |    |    |    |    |    |    |    |      |    |         |    |    |    |    |    |
|-------|----|----|----|----|----|----|----|----|------|----|---------|----|----|----|----|----|
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24   | 25 | 26      | 27 | 28 | 29 | 30 | 31 |
| R     | 0  |    |    |    |    |    |    |    | INVC |    | CHKINVM |    |    |    |    |    |
| W     |    |    |    |    |    |    |    |    |      |    |         |    |    |    |    |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0  | 0       | 0  | 0  | 0  | 0  | 0  |

### SIU\_ECCEIR field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                    |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–23<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                        |
| 24<br>INVC       | Invert Control. Controls whether the ECC checkbits can be inverted during memory read operations on the MCAN memory array.<br><br>0 No checkbit substitution is performed on MCAN memory read accesses.<br>1 Checkbit substitution is performed on MCAN write operations onto the MCAN memory. |
| 25–31<br>CHKINVM | Check Bit Invert Mask. This field specifies which bits of the ECC checkbits are inverted during read operations on the MCAN memory array. For each bit of this field:<br><br>0 The bit is not inverted.<br>1 The bit is inverted.                                                              |

## 8.2.47 Parallel GPIO Pin Data Output Registers (SIU\_PGPDO<sub>n</sub>)

The PGPDO<sub>x</sub> registers are written to by software to drive data out on external GPIO pins. These registers access the same GPIO pins accessed by the SIU\_GPDO0 - SIU\_GPDO511 registers, and they share the same hardware registers. For example, SIU\_PGPDO0 bit 0 is the same as SIU\_GPDO0 bit 7, SIU\_PGPDO0 bit 1 is the same as SIU\_GPDO1 bit 7, and so on.

The parallel GPIO read/write addresses map to the same physical addresses of the normal GPIO. This way both GPDO and corresponding PGPDO registers will be updated by a single write to either registers.

Address: 0h base + C00h offset + (4d × i), where i=0d to 15d

|       |        |        |        |        |        |        |        |        |        |        |         |         |         |         |         |         |
|-------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|---------|---------|---------|---------|---------|---------|
| Bit   | 0      | 1      | 2      | 3      | 4      | 5      | 6      | 7      | 8      | 9      | 10      | 11      | 12      | 13      | 14      | 15      |
| R     | PGPDO0 | PGPDO1 | PGPDO2 | PGPDO3 | PGPDO4 | PGPDO5 | PGPDO6 | PGPDO7 | PGPDO8 | PGPDO9 | PGPDO10 | PGPDO11 | PGPDO12 | PGPDO13 | PGPDO14 | PGPDO15 |
| W     |        |        |        |        |        |        |        |        |        |        |         |         |         |         |         |         |
| Reset | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0       | 0       | 0       | 0       | 0       | 0       |

|       |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
|-------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| Bit   | 16      | 17      | 18      | 19      | 20      | 21      | 22      | 23      | 24      | 25      | 26      | 27      | 28      | 29      | 30      | 31      |
| R     | PGPDO16 | PGPDO17 | PGPDO18 | PGPDO19 | PGPDO20 | PGPDO21 | PGPDO22 | PGPDO23 | PGPDO24 | PGPDO25 | PGPDO26 | PGPDO27 | PGPDO28 | PGPDO29 | PGPDO30 | PGPDO31 |
| W     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| Reset | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       |

### SIU\_PGPDO<sub>n</sub> field descriptions

| Field       | Description                                                                                                                                                                                                                                                                                    |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>PGPDO0 | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br>0 Logic low value is driven on the GPIO pin when the pin is configured as an output.<br>1 Logic high value is driven on the GPIO pin when the pin is configured as an output. |
| 1<br>PGPDO1 | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br>0 Logic low value is driven on the GPIO pin when the pin is configured as an output.<br>1 Logic high value is driven on the GPIO pin when the pin is configured as an output. |
| 2<br>PGPDO2 | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br>0 Logic low value is driven on the GPIO pin when the pin is configured as an output.<br>1 Logic high value is driven on the GPIO pin when the pin is configured as an output. |
| 3<br>PGPDO3 | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br>0 Logic low value is driven on the GPIO pin when the pin is configured as an output.<br>1 Logic high value is driven on the GPIO pin when the pin is configured as an output. |
| 4<br>PGPDO4 | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br>0 Logic low value is driven on the GPIO pin when the pin is configured as an output.<br>1 Logic high value is driven on the GPIO pin when the pin is configured as an output. |
| 5<br>PGPDO5 | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br>0 Logic low value is driven on the GPIO pin when the pin is configured as an output.<br>1 Logic high value is driven on the GPIO pin when the pin is configured as an output. |
| 6<br>PGPDO6 | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br>0 Logic low value is driven on the GPIO pin when the pin is configured as an output.<br>1 Logic high value is driven on the GPIO pin when the pin is configured as an output. |
| 7<br>PGPDO7 | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br>0 Logic low value is driven on the GPIO pin when the pin is configured as an output.<br>1 Logic high value is driven on the GPIO pin when the pin is configured as an output. |
| 8<br>PGPDO8 | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br>0 Logic low value is driven on the GPIO pin when the pin is configured as an output.<br>1 Logic high value is driven on the GPIO pin when the pin is configured as an output. |
| 9<br>PGPDO9 | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br>0 Logic low value is driven on the GPIO pin when the pin is configured as an output.<br>1 Logic high value is driven on the GPIO pin when the pin is configured as an output. |

Table continues on the next page...

**SIU\_PGPDOn field descriptions (continued)**

| Field         | Description                                                                                                                                                                                                                                                                                    |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 10<br>PGPDO10 | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br>0 Logic low value is driven on the GPIO pin when the pin is configured as an output.<br>1 Logic high value is driven on the GPIO pin when the pin is configured as an output. |
| 11<br>PGPDO11 | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br>0 Logic low value is driven on the GPIO pin when the pin is configured as an output.<br>1 Logic high value is driven on the GPIO pin when the pin is configured as an output. |
| 12<br>PGPDO12 | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br>0 Logic low value is driven on the GPIO pin when the pin is configured as an output.<br>1 Logic high value is driven on the GPIO pin when the pin is configured as an output. |
| 13<br>PGPDO13 | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br>0 Logic low value is driven on the GPIO pin when the pin is configured as an output.<br>1 Logic high value is driven on the GPIO pin when the pin is configured as an output. |
| 14<br>PGPDO14 | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br>0 Logic low value is driven on the GPIO pin when the pin is configured as an output.<br>1 Logic high value is driven on the GPIO pin when the pin is configured as an output. |
| 15<br>PGPDO15 | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br>0 Logic low value is driven on the GPIO pin when the pin is configured as an output.<br>1 Logic high value is driven on the GPIO pin when the pin is configured as an output. |
| 16<br>PGPDO16 | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br>0 Logic low value is driven on the GPIO pin when the pin is configured as an output.<br>1 Logic high value is driven on the GPIO pin when the pin is configured as an output. |
| 17<br>PGPDO17 | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br>0 Logic low value is driven on the GPIO pin when the pin is configured as an output.<br>1 Logic high value is driven on the GPIO pin when the pin is configured as an output. |
| 18<br>PGPDO18 | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br>0 Logic low value is driven on the GPIO pin when the pin is configured as an output.<br>1 Logic high value is driven on the GPIO pin when the pin is configured as an output. |
| 19<br>PGPDO19 | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br>0 Logic low value is driven on the GPIO pin when the pin is configured as an output.<br>1 Logic high value is driven on the GPIO pin when the pin is configured as an output. |
| 20<br>PGPDO20 | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br>0 Logic low value is driven on the GPIO pin when the pin is configured as an output.<br>1 Logic high value is driven on the GPIO pin when the pin is configured as an output. |
| 21<br>PGPDO21 | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br>0 Logic low value is driven on the GPIO pin when the pin is configured as an output.<br>1 Logic high value is driven on the GPIO pin when the pin is configured as an output. |

*Table continues on the next page...*



**SIU\_PGPDOn field descriptions (continued)**

| Field         | Description                                                                                                                                                                                                                                                                                    |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 22<br>PGPDO22 | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br>0 Logic low value is driven on the GPIO pin when the pin is configured as an output.<br>1 Logic high value is driven on the GPIO pin when the pin is configured as an output. |
| 23<br>PGPDO23 | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br>0 Logic low value is driven on the GPIO pin when the pin is configured as an output.<br>1 Logic high value is driven on the GPIO pin when the pin is configured as an output. |
| 24<br>PGPDO24 | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br>0 Logic low value is driven on the GPIO pin when the pin is configured as an output.<br>1 Logic high value is driven on the GPIO pin when the pin is configured as an output. |
| 25<br>PGPDO25 | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br>0 Logic low value is driven on the GPIO pin when the pin is configured as an output.<br>1 Logic high value is driven on the GPIO pin when the pin is configured as an output. |
| 26<br>PGPDO26 | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br>0 Logic low value is driven on the GPIO pin when the pin is configured as an output.<br>1 Logic high value is driven on the GPIO pin when the pin is configured as an output. |
| 27<br>PGPDO27 | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br>0 Logic low value is driven on the GPIO pin when the pin is configured as an output.<br>1 Logic high value is driven on the GPIO pin when the pin is configured as an output. |
| 28<br>PGPDO28 | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br>0 Logic low value is driven on the GPIO pin when the pin is configured as an output.<br>1 Logic high value is driven on the GPIO pin when the pin is configured as an output. |
| 29<br>PGPDO29 | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br>0 Logic low value is driven on the GPIO pin when the pin is configured as an output.<br>1 Logic high value is driven on the GPIO pin when the pin is configured as an output. |
| 30<br>PGPDO30 | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br>0 Logic low value is driven on the GPIO pin when the pin is configured as an output.<br>1 Logic high value is driven on the GPIO pin when the pin is configured as an output. |
| 31<br>PGPDO31 | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br>0 Logic low value is driven on the GPIO pin when the pin is configured as an output.<br>1 Logic high value is driven on the GPIO pin when the pin is configured as an output. |

## 8.2.48 Parallel GPIO Pin Data Input Registers (SIU\_PGPDIn)

The GPD<sub>I</sub>x registers are read-only registers that allow reading of external GPIO pins. These registers access the same GPIO pins accessed by the SIU\_GPD<sub>I</sub>0 - SIU\_GPD<sub>I</sub>15 registers, and they share the same hardware registers. For example, SIU\_PGPD<sub>I</sub>0 bit 0 is the same as SIU\_GPD<sub>I</sub>0 bit 7, SIU\_PGPD<sub>I</sub>0 bit 1 is the same as SIU\_GPD<sub>I</sub>1 bit 7, and so on.

The parallel GPIO read/write addresses map to the same physical addresses of the normal GPIO. This way both the GPD<sub>I</sub> and corresponding PGPD<sub>I</sub> registers are updated when a pin state changes, provided that the IBE bit is asserted in the corresponding PCR register.

Address: 0h base + C40h offset + (4d × i), where i=0d to 15d

| Bit   | 0                    | 1                    | 2                    | 3                    | 4                    | 5                    | 6                    | 7                    | 8                    | 9                    | 10                   | 11                   | 12                   | 13                   | 14                   | 15                   |
|-------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|
| R     | PGPD <sub>I</sub> 0  | PGPD <sub>I</sub> 1  | PGPD <sub>I</sub> 2  | PGPD <sub>I</sub> 3  | PGPD <sub>I</sub> 4  | PGPD <sub>I</sub> 5  | PGPD <sub>I</sub> 6  | PGPD <sub>I</sub> 7  | PGPD <sub>I</sub> 8  | PGPD <sub>I</sub> 9  | PGPD <sub>I</sub> 10 | PGPD <sub>I</sub> 11 | PGPD <sub>I</sub> 12 | PGPD <sub>I</sub> 13 | PGPD <sub>I</sub> 14 | PGPD <sub>I</sub> 15 |
| W     |                      |                      |                      |                      |                      |                      |                      |                      |                      |                      |                      |                      |                      |                      |                      |                      |
| Reset | 0                    | 0                    | 0                    | 0                    | 0                    | 0                    | 0                    | 0                    | 0                    | 0                    | 0                    | 0                    | 0                    | 0                    | 0                    | 0                    |
| Bit   | 16                   | 17                   | 18                   | 19                   | 20                   | 21                   | 22                   | 23                   | 24                   | 25                   | 26                   | 27                   | 28                   | 29                   | 30                   | 31                   |
| R     | PGPD <sub>I</sub> 16 | PGPD <sub>I</sub> 17 | PGPD <sub>I</sub> 18 | PGPD <sub>I</sub> 19 | PGPD <sub>I</sub> 20 | PGPD <sub>I</sub> 21 | PGPD <sub>I</sub> 22 | PGPD <sub>I</sub> 23 | PGPD <sub>I</sub> 24 | PGPD <sub>I</sub> 25 | PGPD <sub>I</sub> 26 | PGPD <sub>I</sub> 27 | PGPD <sub>I</sub> 28 | PGPD <sub>I</sub> 29 | PGPD <sub>I</sub> 30 | PGPD <sub>I</sub> 31 |
| W     |                      |                      |                      |                      |                      |                      |                      |                      |                      |                      |                      |                      |                      |                      |                      |                      |
| Reset | 0                    | 0                    | 0                    | 0                    | 0                    | 0                    | 0                    | 0                    | 0                    | 0                    | 0                    | 0                    | 0                    | 0                    | 0                    | 0                    |

**SIU\_PGPDIn field descriptions**

| <b>Field</b> | <b>Description</b>                                                                                                                                                                                                                                                      |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>PGPD10  | Pin Data In. This bit reflects the value of the external GPIO pin associated with the register, provided that the IBE bit is asserted in the corresponding PCR register.<br><br>0 The value of the GPIO pin is logic low.<br>1 The value of the GPIO pin is logic high. |
| 1<br>PGPD11  | Pin Data In. This bit reflects the value of the external GPIO pin associated with the register, provided that the IBE bit is asserted in the corresponding PCR register.<br><br>0 The value of the GPIO pin is logic low.<br>1 The value of the GPIO pin is logic high. |
| 2<br>PGPD12  | Pin Data In. This bit reflects the value of the external GPIO pin associated with the register, provided that the IBE bit is asserted in the corresponding PCR register.<br><br>0 The value of the GPIO pin is logic low.<br>1 The value of the GPIO pin is logic high. |
| 3<br>PGPD13  | Pin Data In. This bit reflects the value of the external GPIO pin associated with the register, provided that the IBE bit is asserted in the corresponding PCR register.<br><br>0 The value of the GPIO pin is logic low.<br>1 The value of the GPIO pin is logic high. |
| 4<br>PGPD14  | Pin Data In. This bit reflects the value of the external GPIO pin associated with the register, provided that the IBE bit is asserted in the corresponding PCR register.<br><br>0 The value of the GPIO pin is logic low.<br>1 The value of the GPIO pin is logic high. |
| 5<br>PGPD15  | Pin Data In. This bit reflects the value of the external GPIO pin associated with the register, provided that the IBE bit is asserted in the corresponding PCR register.<br><br>0 The value of the GPIO pin is logic low.<br>1 The value of the GPIO pin is logic high. |
| 6<br>PGPD16  | Pin Data In. This bit reflects the value of the external GPIO pin associated with the register, provided that the IBE bit is asserted in the corresponding PCR register.<br><br>0 The value of the GPIO pin is logic low.<br>1 The value of the GPIO pin is logic high. |
| 7<br>PGPD17  | Pin Data In. This bit reflects the value of the external GPIO pin associated with the register, provided that the IBE bit is asserted in the corresponding PCR register.<br><br>0 The value of the GPIO pin is logic low.<br>1 The value of the GPIO pin is logic high. |
| 8<br>PGPD18  | Pin Data In. This bit reflects the value of the external GPIO pin associated with the register, provided that the IBE bit is asserted in the corresponding PCR register.<br><br>0 The value of the GPIO pin is logic low.<br>1 The value of the GPIO pin is logic high. |
| 9<br>PGPD19  | Pin Data In. This bit reflects the value of the external GPIO pin associated with the register, provided that the IBE bit is asserted in the corresponding PCR register.<br><br>0 The value of the GPIO pin is logic low.<br>1 The value of the GPIO pin is logic high. |

*Table continues on the next page...*

**SIU\_PGPDI<sub>n</sub> field descriptions (continued)**

| Field         | Description                                                                                                                                                                                                                                                             |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 10<br>PGPDI10 | Pin Data In. This bit reflects the value of the external GPIO pin associated with the register, provided that the IBE bit is asserted in the corresponding PCR register.<br><br>0 The value of the GPIO pin is logic low.<br>1 The value of the GPIO pin is logic high. |
| 11<br>PGPDI11 | Pin Data In. This bit reflects the value of the external GPIO pin associated with the register, provided that the IBE bit is asserted in the corresponding PCR register.<br><br>0 The value of the GPIO pin is logic low.<br>1 The value of the GPIO pin is logic high. |
| 12<br>PGPDI12 | Pin Data In. This bit reflects the value of the external GPIO pin associated with the register, provided that the IBE bit is asserted in the corresponding PCR register.<br><br>0 The value of the GPIO pin is logic low.<br>1 The value of the GPIO pin is logic high. |
| 13<br>PGPDI13 | Pin Data In. This bit reflects the value of the external GPIO pin associated with the register, provided that the IBE bit is asserted in the corresponding PCR register.<br><br>0 The value of the GPIO pin is logic low.<br>1 The value of the GPIO pin is logic high. |
| 14<br>PGPDI14 | Pin Data In. This bit reflects the value of the external GPIO pin associated with the register, provided that the IBE bit is asserted in the corresponding PCR register.<br><br>0 The value of the GPIO pin is logic low.<br>1 The value of the GPIO pin is logic high. |
| 15<br>PGPDI15 | Pin Data In. This bit reflects the value of the external GPIO pin associated with the register, provided that the IBE bit is asserted in the corresponding PCR register.<br><br>0 The value of the GPIO pin is logic low.<br>1 The value of the GPIO pin is logic high. |
| 16<br>PGPDI16 | Pin Data In. This bit reflects the value of the external GPIO pin associated with the register, provided that the IBE bit is asserted in the corresponding PCR register.<br><br>0 The value of the GPIO pin is logic low.<br>1 The value of the GPIO pin is logic high. |
| 17<br>PGPDI17 | Pin Data In. This bit reflects the value of the external GPIO pin associated with the register, provided that the IBE bit is asserted in the corresponding PCR register.<br><br>0 The value of the GPIO pin is logic low.<br>1 The value of the GPIO pin is logic high. |
| 18<br>PGPDI18 | Pin Data In. This bit reflects the value of the external GPIO pin associated with the register, provided that the IBE bit is asserted in the corresponding PCR register.<br><br>0 The value of the GPIO pin is logic low.<br>1 The value of the GPIO pin is logic high. |
| 19<br>PGPDI19 | Pin Data In. This bit reflects the value of the external GPIO pin associated with the register, provided that the IBE bit is asserted in the corresponding PCR register.<br><br>0 The value of the GPIO pin is logic low.<br>1 The value of the GPIO pin is logic high. |

*Table continues on the next page...*

**SIU\_PGPDI<sub>n</sub> field descriptions (continued)**

| Field         | Description                                                                                                                                                                                                                                                             |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 20<br>PGPDI20 | Pin Data In. This bit reflects the value of the external GPIO pin associated with the register, provided that the IBE bit is asserted in the corresponding PCR register.<br><br>0 The value of the GPIO pin is logic low.<br>1 The value of the GPIO pin is logic high. |
| 21<br>PGPDI21 | Pin Data In. This bit reflects the value of the external GPIO pin associated with the register, provided that the IBE bit is asserted in the corresponding PCR register.<br><br>0 The value of the GPIO pin is logic low.<br>1 The value of the GPIO pin is logic high. |
| 22<br>PGPDI22 | Pin Data In. This bit reflects the value of the external GPIO pin associated with the register, provided that the IBE bit is asserted in the corresponding PCR register.<br><br>0 The value of the GPIO pin is logic low.<br>1 The value of the GPIO pin is logic high. |
| 23<br>PGPDI23 | Pin Data In. This bit reflects the value of the external GPIO pin associated with the register, provided that the IBE bit is asserted in the corresponding PCR register.<br><br>0 The value of the GPIO pin is logic low.<br>1 The value of the GPIO pin is logic high. |
| 24<br>PGPDI24 | Pin Data In. This bit reflects the value of the external GPIO pin associated with the register, provided that the IBE bit is asserted in the corresponding PCR register.<br><br>0 The value of the GPIO pin is logic low.<br>1 The value of the GPIO pin is logic high. |
| 25<br>PGPDI25 | Pin Data In. This bit reflects the value of the external GPIO pin associated with the register, provided that the IBE bit is asserted in the corresponding PCR register.<br><br>0 The value of the GPIO pin is logic low.<br>1 The value of the GPIO pin is logic high. |
| 26<br>PGPDI26 | Pin Data In. This bit reflects the value of the external GPIO pin associated with the register, provided that the IBE bit is asserted in the corresponding PCR register.<br><br>0 The value of the GPIO pin is logic low.<br>1 The value of the GPIO pin is logic high. |
| 27<br>PGPDI27 | Pin Data In. This bit reflects the value of the external GPIO pin associated with the register, provided that the IBE bit is asserted in the corresponding PCR register.<br><br>0 The value of the GPIO pin is logic low.<br>1 The value of the GPIO pin is logic high. |
| 28<br>PGPDI28 | Pin Data In. This bit reflects the value of the external GPIO pin associated with the register, provided that the IBE bit is asserted in the corresponding PCR register.<br><br>0 The value of the GPIO pin is logic low.<br>1 The value of the GPIO pin is logic high. |
| 29<br>PGPDI29 | Pin Data In. This bit reflects the value of the external GPIO pin associated with the register, provided that the IBE bit is asserted in the corresponding PCR register.<br><br>0 The value of the GPIO pin is logic low.<br>1 The value of the GPIO pin is logic high. |

*Table continues on the next page...*

**SIU\_PGPDIn field descriptions (continued)**

| Field         | Description                                                                                                                                                                                                                                                             |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 30<br>PGPDI30 | Pin Data In. This bit reflects the value of the external GPIO pin associated with the register, provided that the IBE bit is asserted in the corresponding PCR register.<br><br>0 The value of the GPIO pin is logic low.<br>1 The value of the GPIO pin is logic high. |
| 31<br>PGPDI31 | Pin Data In. This bit reflects the value of the external GPIO pin associated with the register, provided that the IBE bit is asserted in the corresponding PCR register.<br><br>0 The value of the GPIO pin is logic low.<br>1 The value of the GPIO pin is logic high. |

## 8.2.49 Masked Parallel GPIO Pin Data Output Registers (SIU\_MPGPDO<sub>n</sub>)

The MPGPDO<sub>n</sub> are write-only registers that are used to drive data out on external GPIO pins. These registers access the same GPIO pins accessed by SIU\_GPDO0 - SIU\_GPDO511 bit registers. The 16 least significant bits in the SIU\_MPGPDO share the same hardware registers used by SIU\_GPDO. For example, SIU\_MPGPDO<sub>n</sub> bit 16 is the same as SIU\_GPDO<sub>n</sub> bit 7, SIU\_MPGPDO0 bit 17 is the same as SIU\_GPDO1 bit 7, and so on.

The mask bits qualify the write operation of the data bits. Writing to DATAn is only effective if the corresponding MASK<sub>n</sub> bit is written to logic one at the same time (same access to the register). The value that was written to SIU\_MPGPDO cannot be read back by a read operation to the same register. It can, however, be accessed by reading the corresponding SIU\_GPDO register.

Address: 0h base + C80h offset + (4d × i), where i=0d to 31d

| Bit   | 0     | 1     | 2     | 3     | 4     | 5     | 6     | 7     | 8     | 9     | 10     | 11     | 12     | 13     | 14     | 15     |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|--------|--------|--------|--------|--------|
| R     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0      | 0      | 0      | 0      | 0      | 0      |
| W     | MASK0 | MASK1 | MASK2 | MASK3 | MASK4 | MASK5 | MASK6 | MASK7 | MASK8 | MASK9 | MASK10 | MASK11 | MASK12 | MASK13 | MASK14 | MASK15 |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0      | 0      | 0      | 0      | 0      | 0      |

|       |       |       |       |       |       |       |       |       |       |       |        |        |        |        |        |        |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|--------|--------|--------|--------|--------|
| Bit   | 16    | 17    | 18    | 19    | 20    | 21    | 22    | 23    | 24    | 25    | 26     | 27     | 28     | 29     | 30     | 31     |
| R     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0      | 0      | 0      | 0      | 0      | 0      |
| W     | DATA0 | DATA1 | DATA2 | DATA3 | DATA4 | DATA5 | DATA6 | DATA7 | DATA8 | DATA9 | DATA10 | DATA11 | DATA12 | DATA13 | DATA14 | DATA15 |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0      | 0      | 0      | 0      | 0      | 0      |

SIU\_MPGPDO<sub>n</sub> field descriptions

| Field      | Description                                                                                                                                                                                     |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>MASK0 | Mask Bit. This bit controls the write access to the corresponding GPDO.<br>0 Previous value defined by GPDO is maintained.<br>1 Corresponding GPDO is written with value defined by DATA field. |
| 1<br>MASK1 | Mask Bit. This bit controls the write access to the corresponding GPDO.<br>0 Previous value defined by GPDO is maintained.<br>1 Corresponding GPDO is written with value defined by DATA field. |
| 2<br>MASK2 | Mask Bit. This bit controls the write access to the corresponding GPDO.<br>0 Previous value defined by GPDO is maintained.<br>1 Corresponding GPDO is written with value defined by DATA field. |
| 3<br>MASK3 | Mask Bit. This bit controls the write access to the corresponding GPDO.<br>0 Previous value defined by GPDO is maintained.<br>1 Corresponding GPDO is written with value defined by DATA field. |
| 4<br>MASK4 | Mask Bit. This bit controls the write access to the corresponding GPDO.<br>0 Previous value defined by GPDO is maintained.<br>1 Corresponding GPDO is written with value defined by DATA field. |
| 5<br>MASK5 | Mask Bit. This bit controls the write access to the corresponding GPDO.<br>0 Previous value defined by GPDO is maintained.<br>1 Corresponding GPDO is written with value defined by DATA field. |
| 6<br>MASK6 | Mask Bit. This bit controls the write access to the corresponding GPDO.<br>0 Previous value defined by GPDO is maintained.<br>1 Corresponding GPDO is written with value defined by DATA field. |
| 7<br>MASK7 | Mask Bit. This bit controls the write access to the corresponding GPDO.<br>0 Previous value defined by GPDO is maintained.<br>1 Corresponding GPDO is written with value defined by DATA field. |
| 8<br>MASK8 | Mask Bit. This bit controls the write access to the corresponding GPDO.<br>0 Previous value defined by GPDO is maintained.<br>1 Corresponding GPDO is written with value defined by DATA field. |

Table continues on the next page...

**SIU\_MPGPDOn field descriptions (continued)**

| Field        | Description                                                                                                                                                                                                                                                                                                    |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 9<br>MASK9   | Mask Bit. This bit controls the write access to the corresponding GPDO.<br>0 Previous value defined by GPDO is maintained.<br>1 Corresponding GPDO is written with value defined by DATA field.                                                                                                                |
| 10<br>MASK10 | Mask Bit. This bit controls the write access to the corresponding GPDO.<br>0 Previous value defined by GPDO is maintained.<br>1 Corresponding GPDO is written with value defined by DATA field.                                                                                                                |
| 11<br>MASK11 | Mask Bit. This bit controls the write access to the corresponding GPDO.<br>0 Previous value defined by GPDO is maintained.<br>1 Corresponding GPDO is written with value defined by DATA field.                                                                                                                |
| 12<br>MASK12 | Mask Bit. This bit controls the write access to the corresponding GPDO.<br>0 Previous value defined by GPDO is maintained.<br>1 Corresponding GPDO is written with value defined by DATA field.                                                                                                                |
| 13<br>MASK13 | Mask Bit. This bit controls the write access to the corresponding GPDO.<br>0 Previous value defined by GPDO is maintained.<br>1 Corresponding GPDO is written with value defined by DATA field.                                                                                                                |
| 14<br>MASK14 | Mask Bit. This bit controls the write access to the corresponding GPDO.<br>0 Previous value defined by GPDO is maintained.<br>1 Corresponding GPDO is written with value defined by DATA field.                                                                                                                |
| 15<br>MASK15 | Mask Bit. This bit controls the write access to the corresponding GPDO.<br>0 Previous value defined by GPDO is maintained.<br>1 Corresponding GPDO is written with value defined by DATA field.                                                                                                                |
| 16<br>DATA0  | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br>0 Logic low is driven on the corresponding GPIO pin when the pin is configured as an output.<br>1 Logic high is driven on the corresponding GPIO pin when the pin is configured as an output. |
| 17<br>DATA1  | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br>0 Logic low is driven on the corresponding GPIO pin when the pin is configured as an output.<br>1 Logic high is driven on the corresponding GPIO pin when the pin is configured as an output. |
| 18<br>DATA2  | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br>0 Logic low is driven on the corresponding GPIO pin when the pin is configured as an output.<br>1 Logic high is driven on the corresponding GPIO pin when the pin is configured as an output. |
| 19<br>DATA3  | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br>0 Logic low is driven on the corresponding GPIO pin when the pin is configured as an output.<br>1 Logic high is driven on the corresponding GPIO pin when the pin is configured as an output. |
| 20<br>DATA4  | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br>0 Logic low is driven on the corresponding GPIO pin when the pin is configured as an output.<br>1 Logic high is driven on the corresponding GPIO pin when the pin is configured as an output. |

*Table continues on the next page...*



**SIU\_MPGPDO<sub>n</sub> field descriptions (continued)**

| Field        | Description                                                                                                                                                                                                                                                                                                        |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 21<br>DATA5  | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br><br>0 Logic low is driven on the corresponding GPIO pin when the pin is configured as an output.<br>1 Logic high is driven on the corresponding GPIO pin when the pin is configured as an output. |
| 22<br>DATA6  | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br><br>0 Logic low is driven on the corresponding GPIO pin when the pin is configured as an output.<br>1 Logic high is driven on the corresponding GPIO pin when the pin is configured as an output. |
| 23<br>DATA7  | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br><br>0 Logic low is driven on the corresponding GPIO pin when the pin is configured as an output.<br>1 Logic high is driven on the corresponding GPIO pin when the pin is configured as an output. |
| 24<br>DATA8  | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br><br>0 Logic low is driven on the corresponding GPIO pin when the pin is configured as an output.<br>1 Logic high is driven on the corresponding GPIO pin when the pin is configured as an output. |
| 25<br>DATA9  | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br><br>0 Logic low is driven on the corresponding GPIO pin when the pin is configured as an output.<br>1 Logic high is driven on the corresponding GPIO pin when the pin is configured as an output. |
| 26<br>DATA10 | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br><br>0 Logic low is driven on the corresponding GPIO pin when the pin is configured as an output.<br>1 Logic high is driven on the corresponding GPIO pin when the pin is configured as an output. |
| 27<br>DATA11 | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br><br>0 Logic low is driven on the corresponding GPIO pin when the pin is configured as an output.<br>1 Logic high is driven on the corresponding GPIO pin when the pin is configured as an output. |
| 28<br>DATA12 | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br><br>0 Logic low is driven on the corresponding GPIO pin when the pin is configured as an output.<br>1 Logic high is driven on the corresponding GPIO pin when the pin is configured as an output. |
| 29<br>DATA13 | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br><br>0 Logic low is driven on the corresponding GPIO pin when the pin is configured as an output.<br>1 Logic high is driven on the corresponding GPIO pin when the pin is configured as an output. |
| 30<br>DATA14 | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br><br>0 Logic low is driven on the corresponding GPIO pin when the pin is configured as an output.<br>1 Logic high is driven on the corresponding GPIO pin when the pin is configured as an output. |
| 31<br>DATA15 | Pin Data Out. This bit stores the data to be driven out on the external GPIO pin controlled by this register.<br><br>0 Logic low is driven on the corresponding GPIO pin when the pin is configured as an output.<br>1 Logic high is driven on the corresponding GPIO pin when the pin is configured as an output. |

### 8.2.50 Mask-Output High Register (SIU\_DSPI<sub>n</sub>H)

Each bit in the DSPI serialized output frame can contain a signal routed from the output of one of three on-chip sources. The sources on this device are the eTPU module, eMIOS module, or a software-updated GPO data register. The mapping between module or data register and the output frame for a single DSPI channel is performed by independent 32-bit registers, one for each module plus one for the data register.

These registers are one of the possible sources for the DSPI serialized output frame. The purpose of these registers is to allow any combination of bits in each half of the 32 bit serialized data frame to be updated with a single 32-bit write operation, while allowing other bits to maintain their previous state. This is accomplished by writing a 16 bit masked value coherently with an update value contained in a 16 bit output field, and only updating those bits in the output register for which the corresponding mask bit is set. The mask bits are write-only. A value written to a MASK bit only takes effect on the corresponding DATA bit on the same write cycle in which it is written. Reading a mask bit always returns zero.

Address: 0h base + D00h offset + (8d × i), where i=0d to 3d

| Bit   | 0     | 1     | 2     | 3     | 4     | 5     | 6     | 7     | 8     | 9     | 10     | 11     | 12     | 13     | 14     | 15     |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|--------|--------|--------|--------|--------|--------|
| R     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0      | 0      | 0      | 0      | 0      | 0      |
| W     | MASK0 | MASK1 | MASK2 | MASK3 | MASK4 | MASK5 | MASK6 | MASK7 | MASK8 | MASK9 | MASK10 | MASK11 | MASK12 | MASK13 | MASK14 | MASK15 |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0      | 0      | 0      | 0      | 0      | 0      |
| Bit   | 16    | 17    | 18    | 19    | 20    | 21    | 22    | 23    | 24    | 25    | 26     | 27     | 28     | 29     | 30     | 31     |
| R     |       |       |       |       |       |       |       |       |       |       |        |        |        |        |        |        |
| W     | DATA0 | DATA1 | DATA2 | DATA3 | DATA4 | DATA5 | DATA6 | DATA7 | DATA8 | DATA9 | DATA10 | DATA11 | DATA12 | DATA13 | DATA14 | DATA15 |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0      | 0      | 0      | 0      | 0      | 0      |

SIU\_DSPI<sub>n</sub>H field descriptions

| Field      | Description                                                                                                                                                                                                                         |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>MASK0 | Mask bit. These write-only bits control the write access to the DSPI GPO bits. Reading these bits returns 0.<br><br>0 Previous DSPI GPO value is kept.<br>1 Corresponding DSPI GPO bit is written with value defined by DATA field. |
| 1<br>MASK1 | Mask bit. These write-only bits control the write access to the DSPI GPO bits. Reading these bits returns 0.<br><br>0 Previous DSPI GPO value is kept.<br>1 Corresponding DSPI GPO bit is written with value defined by DATA field. |
| 2<br>MASK2 | Mask bit. These write-only bits control the write access to the DSPI GPO bits. Reading these bits returns 0.<br><br>0 Previous DSPI GPO value is kept.<br>1 Corresponding DSPI GPO bit is written with value defined by DATA field. |
| 3<br>MASK3 | Mask bit. These write-only bits control the write access to the DSPI GPO bits. Reading these bits returns 0.<br><br>0 Previous DSPI GPO value is kept.<br>1 Corresponding DSPI GPO bit is written with value defined by DATA field. |
| 4<br>MASK4 | Mask bit. These write-only bits control the write access to the DSPI GPO bits. Reading these bits returns 0.<br><br>0 Previous DSPI GPO value is kept.<br>1 Corresponding DSPI GPO bit is written with value defined by DATA field. |
| 5<br>MASK5 | Mask bit. These write-only bits control the write access to the DSPI GPO bits. Reading these bits returns 0.<br><br>0 Previous DSPI GPO value is kept.<br>1 Corresponding DSPI GPO bit is written with value defined by DATA field. |
| 6<br>MASK6 | Mask bit. These write-only bits control the write access to the DSPI GPO bits. Reading these bits returns 0.<br><br>0 Previous DSPI GPO value is kept.<br>1 Corresponding DSPI GPO bit is written with value defined by DATA field. |
| 7<br>MASK7 | Mask bit. These write-only bits control the write access to the DSPI GPO bits. Reading these bits returns 0.<br><br>0 Previous DSPI GPO value is kept.<br>1 Corresponding DSPI GPO bit is written with value defined by DATA field. |
| 8<br>MASK8 | Mask bit. These write-only bits control the write access to the DSPI GPO bits. Reading these bits returns 0.<br><br>0 Previous DSPI GPO value is kept.<br>1 Corresponding DSPI GPO bit is written with value defined by DATA field. |
| 9<br>MASK9 | Mask bit. These write-only bits control the write access to the DSPI GPO bits. Reading these bits returns 0.<br><br>0 Previous DSPI GPO value is kept.<br>1 Corresponding DSPI GPO bit is written with value defined by DATA field. |

*Table continues on the next page...*

**SIU\_DSPI<sub>n</sub>H field descriptions (continued)**

| Field        | Description                                                                                                                                                                                                                                                                                                                                                                                                                 |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 10<br>MASK10 | Mask bit. These write-only bits control the write access to the DSPI GPO bits. Reading these bits returns 0.<br><br>0 Previous DSPI GPO value is kept.<br>1 Corresponding DSPI GPO bit is written with value defined by DATA field.                                                                                                                                                                                         |
| 11<br>MASK11 | Mask bit. These write-only bits control the write access to the DSPI GPO bits. Reading these bits returns 0.<br><br>0 Previous DSPI GPO value is kept.<br>1 Corresponding DSPI GPO bit is written with value defined by DATA field.                                                                                                                                                                                         |
| 12<br>MASK12 | Mask bit. These write-only bits control the write access to the DSPI GPO bits. Reading these bits returns 0.<br><br>0 Previous DSPI GPO value is kept.<br>1 Corresponding DSPI GPO bit is written with value defined by DATA field.                                                                                                                                                                                         |
| 13<br>MASK13 | Mask bit. These write-only bits control the write access to the DSPI GPO bits. Reading these bits returns 0.<br><br>0 Previous DSPI GPO value is kept.<br>1 Corresponding DSPI GPO bit is written with value defined by DATA field.                                                                                                                                                                                         |
| 14<br>MASK14 | Mask bit. These write-only bits control the write access to the DSPI GPO bits. Reading these bits returns 0.<br><br>0 Previous DSPI GPO value is kept.<br>1 Corresponding DSPI GPO bit is written with value defined by DATA field.                                                                                                                                                                                         |
| 15<br>MASK15 | Mask bit. These write-only bits control the write access to the DSPI GPO bits. Reading these bits returns 0.<br><br>0 Previous DSPI GPO value is kept.<br>1 Corresponding DSPI GPO bit is written with value defined by DATA field.                                                                                                                                                                                         |
| 16<br>DATA0  | Pin Data Out. Stores the data to be driven out on the GPO for DSPI output controlled by this register.<br><br>0 Logic low value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLA/SIU_DSPIDHLD.<br>1 Logic high value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLA/SIU_DSPIDHLD. |
| 17<br>DATA1  | Pin Data Out. Stores the data to be driven out on the GPO for DSPI output controlled by this register.<br><br>0 Logic low value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLA/SIU_DSPIDHLD.<br>1 Logic high value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLA/SIU_DSPIDHLD. |
| 18<br>DATA2  | Pin Data Out. Stores the data to be driven out on the GPO for DSPI output controlled by this register.<br><br>0 Logic low value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLA/SIU_DSPIDHLD.<br>1 Logic high value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLA/SIU_DSPIDHLD. |
| 19<br>DATA3  | Pin Data Out. Stores the data to be driven out on the GPO for DSPI output controlled by this register.                                                                                                                                                                                                                                                                                                                      |

*Table continues on the next page...*

**SIU\_DSPIaH field descriptions (continued)**

| Field        | Description                                                                                                                                                                                                                                                                                                                                                                                                             |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|              | 0 Logic low value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD.<br>1 Logic high value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD.                                                                                                           |
| 20<br>DATA4  | Pin Data Out. Stores the data to be driven out on the GPO for DSPI output controlled by this register.<br>0 Logic low value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD.<br>1 Logic high value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD. |
| 21<br>DATA5  | Pin Data Out. Stores the data to be driven out on the GPO for DSPI output controlled by this register.<br>0 Logic low value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD.<br>1 Logic high value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD. |
| 22<br>DATA6  | Pin Data Out. Stores the data to be driven out on the GPO for DSPI output controlled by this register.<br>0 Logic low value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD.<br>1 Logic high value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD. |
| 23<br>DATA7  | Pin Data Out. Stores the data to be driven out on the GPO for DSPI output controlled by this register.<br>0 Logic low value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD.<br>1 Logic high value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD. |
| 24<br>DATA8  | Pin Data Out. Stores the data to be driven out on the GPO for DSPI output controlled by this register.<br>0 Logic low value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD.<br>1 Logic high value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD. |
| 25<br>DATA9  | Pin Data Out. Stores the data to be driven out on the GPO for DSPI output controlled by this register.<br>0 Logic low value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD.<br>1 Logic high value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD. |
| 26<br>DATA10 | Pin Data Out. Stores the data to be driven out on the GPO for DSPI output controlled by this register.<br>0 Logic low value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD.<br>1 Logic high value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD. |
| 27<br>DATA11 | Pin Data Out. Stores the data to be driven out on the GPO for DSPI output controlled by this register.                                                                                                                                                                                                                                                                                                                  |

*Table continues on the next page...*

**SIU\_DSPIInH field descriptions (continued)**

| Field        | Description                                                                                                                                                                                                                                                                                                                                                                                                             |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|              | 0 Logic low value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD.<br>1 Logic high value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD.                                                                                                           |
| 28<br>DATA12 | Pin Data Out. Stores the data to be driven out on the GPO for DSPI output controlled by this register.<br>0 Logic low value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD.<br>1 Logic high value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD. |
| 29<br>DATA13 | Pin Data Out. Stores the data to be driven out on the GPO for DSPI output controlled by this register.<br>0 Logic low value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD.<br>1 Logic high value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD. |
| 30<br>DATA14 | Pin Data Out. Stores the data to be driven out on the GPO for DSPI output controlled by this register.<br>0 Logic low value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD.<br>1 Logic high value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD. |
| 31<br>DATA15 | Pin Data Out. Stores the data to be driven out on the GPO for DSPI output controlled by this register.<br>0 Logic low value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD.<br>1 Logic high value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD. |

**8.2.51 Mask-Output Low Register (SIU\_DSPIInL)**

Each bit in the DSPI serialized output frame can contain a signal routed from the output of one of three on-chip sources. The sources on this device are the eTPU module, eMIOS module, or a software-updated GPO data register. The mapping between module or data register and the output frame for a single DSPI channel is performed by independent 32-bit registers, one for each module plus one for the data register.

These registers are one of the possible sources for the DSPI serialized output frame. The purpose of these registers is to allow any combination of bits in each half of the 32 bit serialized data frame to be updated with a single 32-bit write operation, while allowing other bits to maintain their previous state. This is accomplished by writing a 16 bit masked value coherently with an update value contained in a 16 bit output field, and only updating those bits in the output register for which the corresponding mask bit is set. The

mask bits are write-only. A value written to a MASK bit only takes effect on the corresponding DATA bit on the same write cycle in which it is written. Reading a mask bit always returns zero.

Address: 0h base + D04h offset + (8d × i), where i=0d to 3d

|       |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
|-------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Bit   | 0      | 1      | 2      | 3      | 4      | 5      | 6      | 7      | 8      | 9      | 10     | 11     | 12     | 13     | 14     | 15     |
| R     | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      |
| W     | MASK16 | MASK17 | MASK18 | MASK19 | MASK20 | MASK21 | MASK22 | MASK23 | MASK24 | MASK25 | MASK26 | MASK27 | MASK28 | MASK29 | MASK30 | MASK31 |
| Reset | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      |

|       |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
|-------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Bit   | 16     | 17     | 18     | 19     | 20     | 21     | 22     | 23     | 24     | 25     | 26     | 27     | 28     | 29     | 30     | 31     |
| R     |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
| W     | DATA16 | DATA17 | DATA18 | DATA19 | DATA20 | DATA21 | DATA22 | DATA23 | DATA24 | DATA25 | DATA26 | DATA27 | DATA28 | DATA29 | DATA30 | DATA31 |
| Reset | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      |

### SIU\_DSPI<sub>n</sub>L field descriptions

| Field       | Description                                                                                                                                                                                                                         |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>MASK16 | Mask bit. These write-only bits control the write access to the DSPI GPO bits. Reading these bits returns 0.<br><br>0 Previous DSPI GPO value is kept.<br>1 Corresponding DSPI GPO bit is written with value defined by DATA field. |
| 1<br>MASK17 | Mask bit. These write-only bits control the write access to the DSPI GPO bits. Reading these bits returns 0.<br><br>0 Previous DSPI GPO value is kept.<br>1 Corresponding DSPI GPO bit is written with value defined by DATA field. |
| 2<br>MASK18 | Mask bit. These write-only bits control the write access to the DSPI GPO bits. Reading these bits returns 0.<br><br>0 Previous DSPI GPO value is kept.<br>1 Corresponding DSPI GPO bit is written with value defined by DATA field. |
| 3<br>MASK19 | Mask bit. These write-only bits control the write access to the DSPI GPO bits. Reading these bits returns 0.<br><br>0 Previous DSPI GPO value is kept.<br>1 Corresponding DSPI GPO bit is written with value defined by DATA field. |

Table continues on the next page...

**SIU\_DSPInL field descriptions (continued)**

| Field        | Description                                                                                                                                                                                                                         |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 4<br>MASK20  | Mask bit. These write-only bits control the write access to the DSPI GPO bits. Reading these bits returns 0.<br><br>0 Previous DSPI GPO value is kept.<br>1 Corresponding DSPI GPO bit is written with value defined by DATA field. |
| 5<br>MASK21  | Mask bit. These write-only bits control the write access to the DSPI GPO bits. Reading these bits returns 0.<br><br>0 Previous DSPI GPO value is kept.<br>1 Corresponding DSPI GPO bit is written with value defined by DATA field. |
| 6<br>MASK22  | Mask bit. These write-only bits control the write access to the DSPI GPO bits. Reading these bits returns 0.<br><br>0 Previous DSPI GPO value is kept.<br>1 Corresponding DSPI GPO bit is written with value defined by DATA field. |
| 7<br>MASK23  | Mask bit. These write-only bits control the write access to the DSPI GPO bits. Reading these bits returns 0.<br><br>0 Previous DSPI GPO value is kept.<br>1 Corresponding DSPI GPO bit is written with value defined by DATA field. |
| 8<br>MASK24  | Mask bit. These write-only bits control the write access to the DSPI GPO bits. Reading these bits returns 0.<br><br>0 Previous DSPI GPO value is kept.<br>1 Corresponding DSPI GPO bit is written with value defined by DATA field. |
| 9<br>MASK25  | Mask bit. These write-only bits control the write access to the DSPI GPO bits. Reading these bits returns 0.<br><br>0 Previous DSPI GPO value is kept.<br>1 Corresponding DSPI GPO bit is written with value defined by DATA field. |
| 10<br>MASK26 | Mask bit. These write-only bits control the write access to the DSPI GPO bits. Reading these bits returns 0.<br><br>0 Previous DSPI GPO value is kept.<br>1 Corresponding DSPI GPO bit is written with value defined by DATA field. |
| 11<br>MASK27 | Mask bit. These write-only bits control the write access to the DSPI GPO bits. Reading these bits returns 0.<br><br>0 Previous DSPI GPO value is kept.<br>1 Corresponding DSPI GPO bit is written with value defined by DATA field. |
| 12<br>MASK28 | Mask bit. These write-only bits control the write access to the DSPI GPO bits. Reading these bits returns 0.<br><br>0 Previous DSPI GPO value is kept.<br>1 Corresponding DSPI GPO bit is written with value defined by DATA field. |
| 13<br>MASK29 | Mask bit. These write-only bits control the write access to the DSPI GPO bits. Reading these bits returns 0.<br><br>0 Previous DSPI GPO value is kept.<br>1 Corresponding DSPI GPO bit is written with value defined by DATA field. |

*Table continues on the next page...*



**SIU\_DSPInL field descriptions (continued)**

| Field        | Description                                                                                                                                                                                                                                                                                                                                                                                                                 |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 14<br>MASK30 | Mask bit. These write-only bits control the write access to the DSPI GPO bits. Reading these bits returns 0.<br><br>0 Previous DSPI GPO value is kept.<br>1 Corresponding DSPI GPO bit is written with value defined by DATA field.                                                                                                                                                                                         |
| 15<br>MASK31 | Mask bit. These write-only bits control the write access to the DSPI GPO bits. Reading these bits returns 0.<br><br>0 Previous DSPI GPO value is kept.<br>1 Corresponding DSPI GPO bit is written with value defined by DATA field.                                                                                                                                                                                         |
| 16<br>DATA16 | Pin Data Out. Stores the data to be driven out on the GPO for DSPI output controlled by this register.<br><br>0 Logic low value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD.<br>1 Logic high value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD. |
| 17<br>DATA17 | Pin Data Out. Stores the data to be driven out on the GPO for DSPI output controlled by this register.<br><br>0 Logic low value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD.<br>1 Logic high value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD. |
| 18<br>DATA18 | Pin Data Out. Stores the data to be driven out on the GPO for DSPI output controlled by this register.<br><br>0 Logic low value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD.<br>1 Logic high value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD. |
| 19<br>DATA19 | Pin Data Out. Stores the data to be driven out on the GPO for DSPI output controlled by this register.<br><br>0 Logic low value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD.<br>1 Logic high value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD. |
| 20<br>DATA20 | Pin Data Out. Stores the data to be driven out on the GPO for DSPI output controlled by this register.<br><br>0 Logic low value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD.<br>1 Logic high value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD. |
| 21<br>DATA21 | Pin Data Out. Stores the data to be driven out on the GPO for DSPI output controlled by this register.<br><br>0 Logic low value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD.<br>1 Logic high value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD. |
| 22<br>DATA22 | Pin Data Out. Stores the data to be driven out on the GPO for DSPI output controlled by this register.                                                                                                                                                                                                                                                                                                                      |

*Table continues on the next page...*

**SIU\_DSPInL field descriptions (continued)**

| Field        | Description                                                                                                                                                                                                                                                                                                                                                                                                             |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|              | 0 Logic low value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD.<br>1 Logic high value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD.                                                                                                           |
| 23<br>DATA23 | Pin Data Out. Stores the data to be driven out on the GPO for DSPI output controlled by this register.<br>0 Logic low value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD.<br>1 Logic high value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD. |
| 24<br>DATA24 | Pin Data Out. Stores the data to be driven out on the GPO for DSPI output controlled by this register.<br>0 Logic low value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD.<br>1 Logic high value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD. |
| 25<br>DATA25 | Pin Data Out. Stores the data to be driven out on the GPO for DSPI output controlled by this register.<br>0 Logic low value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD.<br>1 Logic high value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD. |
| 26<br>DATA26 | Pin Data Out. Stores the data to be driven out on the GPO for DSPI output controlled by this register.<br>0 Logic low value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD.<br>1 Logic high value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD. |
| 27<br>DATA27 | Pin Data Out. Stores the data to be driven out on the GPO for DSPI output controlled by this register.<br>0 Logic low value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD.<br>1 Logic high value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD. |
| 28<br>DATA28 | Pin Data Out. Stores the data to be driven out on the GPO for DSPI output controlled by this register.<br>0 Logic low value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD.<br>1 Logic high value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD. |
| 29<br>DATA29 | Pin Data Out. Stores the data to be driven out on the GPO for DSPI output controlled by this register.<br>0 Logic low value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD.<br>1 Logic high value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD. |
| 30<br>DATA30 | Pin Data Out. Stores the data to be driven out on the GPO for DSPI output controlled by this register.                                                                                                                                                                                                                                                                                                                  |

*Table continues on the next page...*

**SIU\_DSPInL field descriptions (continued)**

| Field        | Description                                                                                                                                                                                                                                                                                                                                                                                                             |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|              | 0 Logic low value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD.<br>1 Logic high value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD.                                                                                                           |
| 31<br>DATA31 | Pin Data Out. Stores the data to be driven out on the GPO for DSPI output controlled by this register.<br>0 Logic low value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD.<br>1 Logic high value is driven to the DSPI serialization interface when this output is selected in SIU_DSPIAHLA/SIU_DSPIBHLB/SIU_DSPICHLC/SIU_DSPIDHLD. |

### 8.2.52 Serialized Output Signal Selection Register for DSPI\_A (SIU\_ETPUBA)

Each bit in the DSPI serialized output frame can contain a signal routed from the output of one of three on-chip sources. The sources on this device are the eTPU module, eMIOS module, or a software-updated GPO data register. The mapping between module or data register and the output frame for a single DSPI channel is performed by independent 32-bit registers, one for each module plus one for the data register.

The following register is used by DSPI\_A to select the sources of the serialized output when running in DSI or CSI configuration.

Each register bit enables a path from the eTPU\_B channel, eMIOS channel and data register bit SIU\_DSPIAH/SIU\_DSPIAL to the equivalent bit position in the DSPI\_A serialized output frame. The user must ensure that bit selections from each of these registers do not overlap. Multiple sources are logically ORed, which provides the potential for combining outputs from multiple timer channels and data registers to produce more complex bit behavior.

Address: 0h base + D40h offset = D40h

|       |         |         |         |         |         |         |        |        |        |        |        |        |        |        |        |        |
|-------|---------|---------|---------|---------|---------|---------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Bit   | 0       | 1       | 2       | 3       | 4       | 5       | 6      | 7      | 8      | 9      | 10     | 11     | 12     | 13     | 14     | 15     |
| R     | ETPUB15 | ETPUB14 | ETPUB13 | ETPUB12 | ETPUB11 | ETPUB10 | ETPUB9 | ETPUB8 | ETPUB7 | ETPUB6 | ETPUB5 | ETPUB4 | ETPUB3 | ETPUB2 | ETPUB1 | ETPUB0 |
| W     | ETPUB15 | ETPUB14 | ETPUB13 | ETPUB12 | ETPUB11 | ETPUB10 | ETPUB9 | ETPUB8 | ETPUB7 | ETPUB6 | ETPUB5 | ETPUB4 | ETPUB3 | ETPUB2 | ETPUB1 | ETPUB0 |
| Reset | 0       | 0       | 0       | 0       | 0       | 0       | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      |

## Memory Map and Register Definition

| Bit   | 16      | 17      | 18      | 19      | 20      | 21      | 22      | 23      | 24      | 25      | 26      | 27      | 28      | 29      | 30      | 31      |
|-------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| R     | ETPUB31 | ETPUB30 | ETPUB29 | ETPUB28 | ETPUB27 | ETPUB26 | ETPUB25 | ETPUB24 | ETPUB23 | ETPUB22 | ETPUB21 | ETPUB20 | ETPUB19 | ETPUB18 | ETPUB17 | ETPUB16 |
| W     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| Reset | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       |

### SIU\_ETPUBA field descriptions

| Field        | Description                                                                                                                                                                                                                                                                         |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>ETPUB15 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_A serialized output frame will use the respective eTPU_B channel. |
| 1<br>ETPUB14 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_A serialized output frame will use the respective eTPU_B channel. |
| 2<br>ETPUB13 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_A serialized output frame will use the respective eTPU_B channel. |
| 3<br>ETPUB12 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_A serialized output frame will use the respective eTPU_B channel. |
| 4<br>ETPUB11 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_A serialized output frame will use the respective eTPU_B channel. |
| 5<br>ETPUB10 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_A serialized output frame will use the respective eTPU_B channel. |
| 6<br>ETPUB9  | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_A serialized output frame will use the respective eTPU_B channel. |
| 7<br>ETPUB8  | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_A serialized output frame will use the respective eTPU_B channel. |
| 8<br>ETPUB7  | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_A serialized output frame will use the respective eTPU_B channel. |
| 9<br>ETPUB6  | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_A serialized output frame will use the respective eTPU_B channel. |

Table continues on the next page...

**SIU\_ETPUBA field descriptions (continued)**

| Field         | Description                                                                                                                                                                                                                                                                         |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 10<br>ETPUB5  | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_A serialized output frame will use the respective eTPU_B channel. |
| 11<br>ETPUB4  | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_A serialized output frame will use the respective eTPU_B channel. |
| 12<br>ETPUB3  | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_A serialized output frame will use the respective eTPU_B channel. |
| 13<br>ETPUB2  | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_A serialized output frame will use the respective eTPU_B channel. |
| 14<br>ETPUB1  | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_A serialized output frame will use the respective eTPU_B channel. |
| 15<br>ETPUB0  | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_A serialized output frame will use the respective eTPU_B channel. |
| 16<br>ETPUB31 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_A serialized output frame will use the respective eTPU_B channel. |
| 17<br>ETPUB30 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_A serialized output frame will use the respective eTPU_B channel. |
| 18<br>ETPUB29 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_A serialized output frame will use the respective eTPU_B channel. |
| 19<br>ETPUB28 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_A serialized output frame will use the respective eTPU_B channel. |
| 20<br>ETPUB27 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_A serialized output frame will use the respective eTPU_B channel. |
| 21<br>ETPUB26 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_A serialized output frame will use the respective eTPU_B channel. |

*Table continues on the next page...*

## SIU\_ETPUBA field descriptions (continued)

| Field         | Description                                                                                                                                                                                                                                                                         |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 22<br>ETPUB25 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_A serialized output frame will use the respective eTPU_B channel. |
| 23<br>ETPUB24 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_A serialized output frame will use the respective eTPU_B channel. |
| 24<br>ETPUB23 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_A serialized output frame will use the respective eTPU_B channel. |
| 25<br>ETPUB22 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_A serialized output frame will use the respective eTPU_B channel. |
| 26<br>ETPUB21 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_A serialized output frame will use the respective eTPU_B channel. |
| 27<br>ETPUB20 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_A serialized output frame will use the respective eTPU_B channel. |
| 28<br>ETPUB19 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_A serialized output frame will use the respective eTPU_B channel. |
| 29<br>ETPUB18 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_A serialized output frame will use the respective eTPU_B channel. |
| 30<br>ETPUB17 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_A serialized output frame will use the respective eTPU_B channel. |
| 31<br>ETPUB16 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_A serialized output frame will use the respective eTPU_B channel. |

### 8.2.53 Serialized Output Signal Selection Register for DSPI\_A (SIU\_EMIOA)

Each bit in the DSPI serialized output frame can contain a signal routed from the output of one of three on-chip sources. The sources on this device are the eTPU module, eMIOS module, or a software-updated GPO data register. The mapping between module or data register and the output frame for a single DSPI channel is performed by independent 32-bit registers, one for each module plus one for the data register.

The following register is used by DSPI\_A to select the sources of the serialized output when running in DSI or CSI configuration.

Each register bit enables a path from the eTPU\_B channel, eMIOS channel and data register bit SIU\_DSPIAH/SIU\_DSPIAL to the equivalent bit position in the DSPI\_A serialized output frame. The user must ensure that bit selections from each of these registers do not overlap. Multiple sources are logically ORed, which provides the potential for combining outputs from multiple timer channels and data registers to produce more complex bit behavior.

Address: 0h base + D44h offset = D44h

| Bit   | 0         | 1         | 2         | 3         | 4         | 5         | 6         | 7         | 8         | 9         | 10        | 11        | 12        | 13        | 14        | 15        |
|-------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| R     | EMIOS0_7_ | EMIOS0_6_ | EMIOS0_5_ | EMIOS0_4_ | EMIOS0_3_ | EMIOS0_2_ | EMIOS0_1_ | EMIOS0_0_ | EMIOS1_0_ | EMIOS1_1_ | EMIOS1_2_ | EMIOS1_3_ | EMIOS1_4_ | EMIOS1_5_ | EMIOS1_6_ | EMIOS1_7_ |
| W     | OUT0      | OUT1      | OUT2      | OUT3      | OUT4      | OUT5      | OUT6      | OUT7      | OUT8      | OUT9      | OUT10     | OUT11     | OUT12     | OUT13     | OUT14     | OUT15     |
| Reset | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         |

| Bit   | 16         | 17         | 18         | 19         | 20         | 21         | 22         | 23         | 24        | 25        | 26        | 27        | 28        | 29        | 30        | 31        |
|-------|------------|------------|------------|------------|------------|------------|------------|------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| R     | EMIOS0_16_ | EMIOS0_17_ | EMIOS0_18_ | EMIOS0_19_ | EMIOS0_20_ | EMIOS0_21_ | EMIOS0_22_ | EMIOS0_23_ | EMIOS0_0_ | EMIOS0_1_ | EMIOS0_2_ | EMIOS0_3_ | EMIOS0_4_ | EMIOS0_5_ | EMIOS0_6_ | EMIOS0_7_ |
| W     | OUT16      | OUT17      | OUT18      | OUT19      | OUT20      | OUT21      | OUT22      | OUT23      | OUT24     | OUT25     | OUT26     | OUT27     | OUT28     | OUT29     | OUT30     | OUT31     |
| Reset | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0          | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         |

**SIU\_EMIOA field descriptions**

| Field                  | Description                                                                                                                                                                                                                                                   |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>EMIOS0_7_<br>OUT0 | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use eMIOS_0 channel 7.<br>1 This bit in the DSPI_A serialized output frame will use eMIOS_0 channel 7. |

*Table continues on the next page...*

## SIU\_EMIOSA field descriptions (continued)

| Field                | Description                                                                                                                                                                                                                                                   |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1<br>EMIOS0_6_OUT1   | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use eMIOS_0 channel 6.<br>1 This bit in the DSPI_A serialized output frame will use eMIOS_0 channel 6. |
| 2<br>EMIOS0_5_OUT2   | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use eMIOS_0 channel 5.<br>1 This bit in the DSPI_A serialized output frame will use eMIOS_0 channel 5. |
| 3<br>EMIOS0_4_OUT3   | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use eMIOS_0 channel 4.<br>1 This bit in the DSPI_A serialized output frame will use eMIOS_0 channel 4. |
| 4<br>EMIOS0_3_OUT4   | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use eMIOS_0 channel 3.<br>1 This bit in the DSPI_A serialized output frame will use eMIOS_0 channel 3. |
| 5<br>EMIOS0_2_OUT5   | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use eMIOS_0 channel 2.<br>1 This bit in the DSPI_A serialized output frame will use eMIOS_0 channel 2. |
| 6<br>EMIOS0_1_OUT6   | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use eMIOS_0 channel 1.<br>1 This bit in the DSPI_A serialized output frame will use eMIOS_0 channel 1. |
| 7<br>EMIOS0_0_OUT7   | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use eMIOS_0 channel 0.<br>1 This bit in the DSPI_A serialized output frame will use eMIOS_0 channel 0. |
| 8<br>EMIOS1_0_OUT8   | eMIOS_1 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use eMIOS_1 channel 0.<br>1 This bit in the DSPI_A serialized output frame will use eMIOS_1 channel 0. |
| 9<br>EMIOS1_1_OUT9   | eMIOS_1 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use eMIOS_1 channel 1.<br>1 This bit in the DSPI_A serialized output frame will use eMIOS_1 channel 1. |
| 10<br>EMIOS1_2_OUT10 | eMIOS_1 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use eMIOS_1 channel 2.<br>1 This bit in the DSPI_A serialized output frame will use eMIOS_1 channel 2. |
| 11<br>EMIOS1_3_OUT11 | eMIOS_1 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use eMIOS_1 channel 3.<br>1 This bit in the DSPI_A serialized output frame will use eMIOS_1 channel 3. |
| 12<br>EMIOS1_4_OUT12 | eMIOS_1 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use eMIOS_1 channel 4.<br>1 This bit in the DSPI_A serialized output frame will use eMIOS_1 channel 4. |

Table continues on the next page...



**SIU\_EMIOA field descriptions (continued)**

| Field                     | Description                                                                                                                                                                                                                                                     |
|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 13<br>EMIOS1_5_<br>OUT13  | eMIOS_1 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use eMIOS_1 channel 5.<br>1 This bit in the DSPI_A serialized output frame will use eMIOS_1 channel 5.   |
| 14<br>EMIOS1_6_<br>OUT14  | eMIOS_1 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use eMIOS_1 channel 6.<br>1 This bit in the DSPI_A serialized output frame will use eMIOS_1 channel 6.   |
| 15<br>EMIOS1_7_<br>OUT15  | eMIOS_1 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use eMIOS_1 channel 7.<br>1 This bit in the DSPI_A serialized output frame will use eMIOS_1 channel 7.   |
| 16<br>EMIOS0_16_<br>OUT16 | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use eMIOS_0 channel 16.<br>1 This bit in the DSPI_A serialized output frame will use eMIOS_0 channel 16. |
| 17<br>EMIOS0_17_<br>OUT17 | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use eMIOS_0 channel 17.<br>1 This bit in the DSPI_A serialized output frame will use eMIOS_0 channel 17. |
| 18<br>EMIOS0_18_<br>OUT18 | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use eMIOS_0 channel 18.<br>1 This bit in the DSPI_A serialized output frame will use eMIOS_0 channel 18. |
| 19<br>EMIOS0_19_<br>OUT19 | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use eMIOS_0 channel 19.<br>1 This bit in the DSPI_A serialized output frame will use eMIOS_0 channel 19. |
| 20<br>EMIOS0_20_<br>OUT20 | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use eMIOS_0 channel 20.<br>1 This bit in the DSPI_A serialized output frame will use eMIOS_0 channel 20. |
| 21<br>EMIOS0_21_<br>OUT21 | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use eMIOS_0 channel 21.<br>1 This bit in the DSPI_A serialized output frame will use eMIOS_0 channel 21. |
| 22<br>EMIOS0_22_<br>OUT22 | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use eMIOS_0 channel 22.<br>1 This bit in the DSPI_A serialized output frame will use eMIOS_0 channel 22. |
| 23<br>EMIOS0_23_<br>OUT23 | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use eMIOS_0 channel 23.<br>1 This bit in the DSPI_A serialized output frame will use eMIOS_0 channel 23. |
| 24<br>EMIOS0_0_<br>OUT24  | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use eMIOS_0 channel 0.<br>1 This bit in the DSPI_A serialized output frame will use eMIOS_0 channel 0.   |

*Table continues on the next page...*

**SIU\_EMIOSA field descriptions (continued)**

| Field                    | Description                                                                                                                                                                                                                                                   |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 25<br>EMIOS0_1_<br>OUT25 | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use eMIOS_0 channel 1.<br>1 This bit in the DSPI_A serialized output frame will use eMIOS_0 channel 1. |
| 26<br>EMIOS0_2_<br>OUT26 | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use eMIOS_0 channel 2.<br>1 This bit in the DSPI_A serialized output frame will use eMIOS_0 channel 2. |
| 27<br>EMIOS0_3_<br>OUT27 | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use eMIOS_0 channel 3.<br>1 This bit in the DSPI_A serialized output frame will use eMIOS_0 channel 3. |
| 28<br>EMIOS0_4_<br>OUT28 | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use eMIOS_0 channel 4.<br>1 This bit in the DSPI_A serialized output frame will use eMIOS_0 channel 4. |
| 29<br>EMIOS0_5_<br>OUT29 | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use eMIOS_0 channel 5.<br>1 This bit in the DSPI_A serialized output frame will use eMIOS_0 channel 5. |
| 30<br>EMIOS0_6_<br>OUT30 | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use eMIOS_0 channel 6.<br>1 This bit in the DSPI_A serialized output frame will use eMIOS_0 channel 6. |
| 31<br>EMIOS0_7_<br>OUT31 | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_A output.<br>0 This bit in the DSPI_A serialized output frame will not use eMIOS_0 channel 7.<br>1 This bit in the DSPI_A serialized output frame will use eMIOS_0 channel 7. |

**8.2.54 Serialized Output Signal Selection Register for DSPI\_A (SIU\_DSPIAHLA)**

Each bit in the DSPI serialized output frame can contain a signal routed from the output of one of three on-chip sources. The sources on this device are the eTPU module, eMIOS module, or a software-updated GPO data register. The mapping between module or data register and the output frame for a single DSPI channel is performed by independent 32-bit registers, one for each module plus one for the data register.

The following register is used by DSPI\_A to select the sources of the serialized output when running in DSI or CSI configuration.

Each register bit enables a path from the eTPU\_B channel, eMIOS channel and data register bit SIU\_DSPIAH/SIU\_DSPIAL to the equivalent bit position in the DSPI\_A serialized output frame. The user must ensure that bit selections from each of these

registers do not overlap. Multiple sources are logically ORed, which provides the potential for combining outputs from multiple timer channels and data registers to produce more complex bit behavior.

Address: 0h base + D48h offset = D48h

|       |         |         |         |         |         |         |         |         |         |         |          |          |          |          |          |          |
|-------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|----------|----------|----------|----------|----------|----------|
| Bit   | 0       | 1       | 2       | 3       | 4       | 5       | 6       | 7       | 8       | 9       | 10       | 11       | 12       | 13       | 14       | 15       |
| R     | DSPIAH0 | DSPIAH1 | DSPIAH2 | DSPIAH3 | DSPIAH4 | DSPIAH5 | DSPIAH6 | DSPIAH7 | DSPIAH8 | DSPIAH9 | DSPIAH10 | DSPIAH11 | DSPIAH12 | DSPIAH13 | DSPIAH14 | DSPIAH15 |
| W     |         |         |         |         |         |         |         |         |         |         |          |          |          |          |          |          |
| Reset | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0        | 0        | 0        | 0        | 0        | 0        |

|       |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |
|-------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| Bit   | 16       | 17       | 18       | 19       | 20       | 21       | 22       | 23       | 24       | 25       | 26       | 27       | 28       | 29       | 30       | 31       |
| R     | DSPIAL16 | DSPIAL17 | DSPIAL18 | DSPIAL19 | DSPIAL20 | DSPIAL21 | DSPIAL22 | DSPIAL23 | DSPIAL24 | DSPIAL25 | DSPIAL26 | DSPIAL27 | DSPIAL28 | DSPIAL29 | DSPIAL30 | DSPIAL31 |
| W     |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |
| Reset | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        |

### SIU\_DSPIAHLA field descriptions

| Field        | Description                                                                                                                                                                                                                                                                                                           |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>DSPIAH0 | SIU_DSPIAH Select. Select one bit of the DSPI_A GPO Mask Output High Register to source the serialized DSPI_A output.<br><br>0 This bit in the DSPI_A serialized output frame will not use the respective SIU_DSPIAH bit.<br>1 This bit in the DSPI_A serialized output frame will use the respective SIU_DSPIAH bit. |
| 1<br>DSPIAH1 | SIU_DSPIAH Select. Select one bit of the DSPI_A GPO Mask Output High Register to source the serialized DSPI_A output.<br><br>0 This bit in the DSPI_A serialized output frame will not use the respective SIU_DSPIAH bit.<br>1 This bit in the DSPI_A serialized output frame will use the respective SIU_DSPIAH bit. |
| 2<br>DSPIAH2 | SIU_DSPIAH Select. Select one bit of the DSPI_A GPO Mask Output High Register to source the serialized DSPI_A output.<br><br>0 This bit in the DSPI_A serialized output frame will not use the respective SIU_DSPIAH bit.<br>1 This bit in the DSPI_A serialized output frame will use the respective SIU_DSPIAH bit. |
| 3<br>DSPIAH3 | SIU_DSPIAH Select. Select one bit of the DSPI_A GPO Mask Output High Register to source the serialized DSPI_A output.<br><br>0 This bit in the DSPI_A serialized output frame will not use the respective SIU_DSPIAH bit.<br>1 This bit in the DSPI_A serialized output frame will use the respective SIU_DSPIAH bit. |
| 4<br>DSPIAH4 | SIU_DSPIAH Select. Select one bit of the DSPI_A GPO Mask Output High Register to source the serialized DSPI_A output.<br><br>0 This bit in the DSPI_A serialized output frame will not use the respective SIU_DSPIAH bit.<br>1 This bit in the DSPI_A serialized output frame will use the respective SIU_DSPIAH bit. |
| 5<br>DSPIAH5 | SIU_DSPIAH Select. Select one bit of the DSPI_A GPO Mask Output High Register to source the serialized DSPI_A output.                                                                                                                                                                                                 |

Table continues on the next page...

**SIU\_DSPIAHLA field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                                                                           |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | 0 This bit in the DSPI_A serialized output frame will not use the respective SIU_DSPIAH bit.<br>1 This bit in the DSPI_A serialized output frame will use the respective SIU_DSPIAH bit.                                                                                                                              |
| 6<br>DSPIAH6   | SIU_DSPIAH Select. Select one bit of the DSPI_A GPO Mask Output High Register to source the serialized DSPI_A output.<br><br>0 This bit in the DSPI_A serialized output frame will not use the respective SIU_DSPIAH bit.<br>1 This bit in the DSPI_A serialized output frame will use the respective SIU_DSPIAH bit. |
| 7<br>DSPIAH7   | SIU_DSPIAH Select. Select one bit of the DSPI_A GPO Mask Output High Register to source the serialized DSPI_A output.<br><br>0 This bit in the DSPI_A serialized output frame will not use the respective SIU_DSPIAH bit.<br>1 This bit in the DSPI_A serialized output frame will use the respective SIU_DSPIAH bit. |
| 8<br>DSPIAH8   | SIU_DSPIAH Select. Select one bit of the DSPI_A GPO Mask Output High Register to source the serialized DSPI_A output.<br><br>0 This bit in the DSPI_A serialized output frame will not use the respective SIU_DSPIAH bit.<br>1 This bit in the DSPI_A serialized output frame will use the respective SIU_DSPIAH bit. |
| 9<br>DSPIAH9   | SIU_DSPIAH Select. Select one bit of the DSPI_A GPO Mask Output High Register to source the serialized DSPI_A output.<br><br>0 This bit in the DSPI_A serialized output frame will not use the respective SIU_DSPIAH bit.<br>1 This bit in the DSPI_A serialized output frame will use the respective SIU_DSPIAH bit. |
| 10<br>DSPIAH10 | SIU_DSPIAH Select. Select one bit of the DSPI_A GPO Mask Output High Register to source the serialized DSPI_A output.<br><br>0 This bit in the DSPI_A serialized output frame will not use the respective SIU_DSPIAH bit.<br>1 This bit in the DSPI_A serialized output frame will use the respective SIU_DSPIAH bit. |
| 11<br>DSPIAH11 | SIU_DSPIAH Select. Select one bit of the DSPI_A GPO Mask Output High Register to source the serialized DSPI_A output.<br><br>0 This bit in the DSPI_A serialized output frame will not use the respective SIU_DSPIAH bit.<br>1 This bit in the DSPI_A serialized output frame will use the respective SIU_DSPIAH bit. |
| 12<br>DSPIAH12 | SIU_DSPIAH Select. Select one bit of the DSPI_A GPO Mask Output High Register to source the serialized DSPI_A output.<br><br>0 This bit in the DSPI_A serialized output frame will not use the respective SIU_DSPIAH bit.<br>1 This bit in the DSPI_A serialized output frame will use the respective SIU_DSPIAH bit. |
| 13<br>DSPIAH13 | SIU_DSPIAH Select. Select one bit of the DSPI_A GPO Mask Output High Register to source the serialized DSPI_A output.<br><br>0 This bit in the DSPI_A serialized output frame will not use the respective SIU_DSPIAH bit.<br>1 This bit in the DSPI_A serialized output frame will use the respective SIU_DSPIAH bit. |
| 14<br>DSPIAH14 | SIU_DSPIAH Select. Select one bit of the DSPI_A GPO Mask Output High Register to source the serialized DSPI_A output.<br><br>0 This bit in the DSPI_A serialized output frame will not use the respective SIU_DSPIAH bit.<br>1 This bit in the DSPI_A serialized output frame will use the respective SIU_DSPIAH bit. |
| 15<br>DSPIAH15 | SIU_DSPIAH Select. Select one bit of the DSPI_A GPO Mask Output High Register to source the serialized DSPI_A output.                                                                                                                                                                                                 |

*Table continues on the next page...*

**SIU\_DSPIAHLA field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                                                                          |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | 0 This bit in the DSPI_A serialized output frame will not use the respective SIU_DSPIAH bit.<br>1 This bit in the DSPI_A serialized output frame will use the respective SIU_DSPIAH bit.                                                                                                                             |
| 16<br>DSPIAL16 | SIU_DSPIAL Select. Select one bit of the DSPI_A GPO Mask Output Low Register to source the serialized DSPI_A output.<br><br>0 This bit in the DSPI_A serialized output frame will not use the respective SIU_DSPIAL bit.<br>1 This bit in the DSPI_A serialized output frame will use the respective SIU_DSPIAL bit. |
| 17<br>DSPIAL17 | SIU_DSPIAL Select. Select one bit of the DSPI_A GPO Mask Output Low Register to source the serialized DSPI_A output.<br><br>0 This bit in the DSPI_A serialized output frame will not use the respective SIU_DSPIAL bit.<br>1 This bit in the DSPI_A serialized output frame will use the respective SIU_DSPIAL bit. |
| 18<br>DSPIAL18 | SIU_DSPIAL Select. Select one bit of the DSPI_A GPO Mask Output Low Register to source the serialized DSPI_A output.<br><br>0 This bit in the DSPI_A serialized output frame will not use the respective SIU_DSPIAL bit.<br>1 This bit in the DSPI_A serialized output frame will use the respective SIU_DSPIAL bit. |
| 19<br>DSPIAL19 | SIU_DSPIAL Select. Select one bit of the DSPI_A GPO Mask Output Low Register to source the serialized DSPI_A output.<br><br>0 This bit in the DSPI_A serialized output frame will not use the respective SIU_DSPIAL bit.<br>1 This bit in the DSPI_A serialized output frame will use the respective SIU_DSPIAL bit. |
| 20<br>DSPIAL20 | SIU_DSPIAL Select. Select one bit of the DSPI_A GPO Mask Output Low Register to source the serialized DSPI_A output.<br><br>0 This bit in the DSPI_A serialized output frame will not use the respective SIU_DSPIAL bit.<br>1 This bit in the DSPI_A serialized output frame will use the respective SIU_DSPIAL bit. |
| 21<br>DSPIAL21 | SIU_DSPIAL Select. Select one bit of the DSPI_A GPO Mask Output Low Register to source the serialized DSPI_A output.<br><br>0 This bit in the DSPI_A serialized output frame will not use the respective SIU_DSPIAL bit.<br>1 This bit in the DSPI_A serialized output frame will use the respective SIU_DSPIAL bit. |
| 22<br>DSPIAL22 | SIU_DSPIAL Select. Select one bit of the DSPI_A GPO Mask Output Low Register to source the serialized DSPI_A output.<br><br>0 This bit in the DSPI_A serialized output frame will not use the respective SIU_DSPIAL bit.<br>1 This bit in the DSPI_A serialized output frame will use the respective SIU_DSPIAL bit. |
| 23<br>DSPIAL23 | SIU_DSPIAL Select. Select one bit of the DSPI_A GPO Mask Output Low Register to source the serialized DSPI_A output.<br><br>0 This bit in the DSPI_A serialized output frame will not use the respective SIU_DSPIAL bit.<br>1 This bit in the DSPI_A serialized output frame will use the respective SIU_DSPIAL bit. |
| 24<br>DSPIAL24 | SIU_DSPIAL Select. Select one bit of the DSPI_A GPO Mask Output Low Register to source the serialized DSPI_A output.<br><br>0 This bit in the DSPI_A serialized output frame will not use the respective SIU_DSPIAL bit.<br>1 This bit in the DSPI_A serialized output frame will use the respective SIU_DSPIAL bit. |
| 25<br>DSPIAL25 | SIU_DSPIAL Select. Select one bit of the DSPI_A GPO Mask Output Low Register to source the serialized DSPI_A output.                                                                                                                                                                                                 |

*Table continues on the next page...*

**SIU\_DSPIAHLA field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                                                                          |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | 0 This bit in the DSPI_A serialized output frame will not use the respective SIU_DSPIAL bit.<br>1 This bit in the DSPI_A serialized output frame will use the respective SIU_DSPIAL bit.                                                                                                                             |
| 26<br>DSPIAL26 | SIU_DSPIAL Select. Select one bit of the DSPI_A GPO Mask Output Low Register to source the serialized DSPI_A output.<br><br>0 This bit in the DSPI_A serialized output frame will not use the respective SIU_DSPIAL bit.<br>1 This bit in the DSPI_A serialized output frame will use the respective SIU_DSPIAL bit. |
| 27<br>DSPIAL27 | SIU_DSPIAL Select. Select one bit of the DSPI_A GPO Mask Output Low Register to source the serialized DSPI_A output.<br><br>0 This bit in the DSPI_A serialized output frame will not use the respective SIU_DSPIAL bit.<br>1 This bit in the DSPI_A serialized output frame will use the respective SIU_DSPIAL bit. |
| 28<br>DSPIAL28 | SIU_DSPIAL Select. Select one bit of the DSPI_A GPO Mask Output Low Register to source the serialized DSPI_A output.<br><br>0 This bit in the DSPI_A serialized output frame will not use the respective SIU_DSPIAL bit.<br>1 This bit in the DSPI_A serialized output frame will use the respective SIU_DSPIAL bit. |
| 29<br>DSPIAL29 | SIU_DSPIAL Select. Select one bit of the DSPI_A GPO Mask Output Low Register to source the serialized DSPI_A output.<br><br>0 This bit in the DSPI_A serialized output frame will not use the respective SIU_DSPIAL bit.<br>1 This bit in the DSPI_A serialized output frame will use the respective SIU_DSPIAL bit. |
| 30<br>DSPIAL30 | SIU_DSPIAL Select. Select one bit of the DSPI_A GPO Mask Output Low Register to source the serialized DSPI_A output.<br><br>0 This bit in the DSPI_A serialized output frame will not use the respective SIU_DSPIAL bit.<br>1 This bit in the DSPI_A serialized output frame will use the respective SIU_DSPIAL bit. |
| 31<br>DSPIAL31 | SIU_DSPIAL Select. Select one bit of the DSPI_A GPO Mask Output Low Register to source the serialized DSPI_A output.<br><br>0 This bit in the DSPI_A serialized output frame will not use the respective SIU_DSPIAL bit.<br>1 This bit in the DSPI_A serialized output frame will use the respective SIU_DSPIAL bit. |

**8.2.55 Serialized Output Signal Selection Register for DSPI\_B (SIU\_ETPUAB)**

Each bit in the DSPI serialized output frame can contain a signal routed from the output of one of three on-chip sources. The sources on this device are the eTPU module, eMIOS module, or a software-updated GPO data register. The mapping between module or data register and the output frame for a single DSPI channel is performed by independent 32-bit registers, one for each module plus one for the data register.

The following register is used by DSPI\_B to select the sources of the serialized output when running in DSI or CSI configuration.

Each register bit enables a path from the eTPU\_A channel, eMIOS channel and data register bit SIU\_DSPIBH/SIU\_DSPIBL to the equivalent bit position in the DSPI\_B serialized output frame. The user must ensure that bit selections from each of these registers do not overlap. Multiple sources are logically ORed, which provides the potential for combining outputs from multiple timer channels and data registers to produce more complex bit behavior.

Address: 0h base + D50h offset = D50h

|       |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
|-------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| Bit   | 0       | 1       | 2       | 3       | 4       | 5       | 6       | 7       | 8       | 9       | 10      | 11      | 12      | 13      | 14      | 15      |
| R     | ETPUA23 | ETPUA22 | ETPUA21 | ETPUA20 | ETPUA19 | ETPUA18 | ETPUA17 | ETPUA16 | ETPUA29 | ETPUA28 | ETPUA27 | ETPUA26 | ETPUA25 | ETPUA24 | ETPUA31 | ETPUA30 |
| W     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| Reset | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       |

|       |         |         |         |         |        |        |        |        |        |        |        |        |        |        |         |         |
|-------|---------|---------|---------|---------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|---------|---------|
| Bit   | 16      | 17      | 18      | 19      | 20     | 21     | 22     | 23     | 24     | 25     | 26     | 27     | 28     | 29     | 30      | 31      |
| R     | ETPUA12 | ETPUA13 | ETPUA14 | ETPUA15 | ETPUA0 | ETPUA1 | ETPUA2 | ETPUA3 | ETPUA4 | ETPUA5 | ETPUA6 | ETPUA7 | ETPUA8 | ETPUA9 | ETPUA10 | ETPUA11 |
| W     |         |         |         |         |        |        |        |        |        |        |        |        |        |        |         |         |
| Reset | 0       | 0       | 0       | 0       | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0       | 0       |

### SIU\_ETPUAB field descriptions

| Field        | Description                                                                                                                                                                                                                                                                         |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>ETPUA23 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_B serialized output frame will use the respective eTPU_A channel. |
| 1<br>ETPUA22 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_B serialized output frame will use the respective eTPU_A channel. |
| 2<br>ETPUA21 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_B serialized output frame will use the respective eTPU_A channel. |
| 3<br>ETPUA20 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_B serialized output frame will use the respective eTPU_A channel. |
| 4<br>ETPUA19 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_B serialized output frame will use the respective eTPU_A channel. |
| 5<br>ETPUA18 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_B serialized output frame will use the respective eTPU_A channel. |

Table continues on the next page...

## SIU\_ETPUAB field descriptions (continued)

| Field         | Description                                                                                                                                                                                                                                                                         |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 6<br>ETPUA17  | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_B serialized output frame will use the respective eTPU_A channel. |
| 7<br>ETPUA16  | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_B serialized output frame will use the respective eTPU_A channel. |
| 8<br>ETPUA29  | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_B serialized output frame will use the respective eTPU_A channel. |
| 9<br>ETPUA28  | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_B serialized output frame will use the respective eTPU_A channel. |
| 10<br>ETPUA27 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_B serialized output frame will use the respective eTPU_A channel. |
| 11<br>ETPUA26 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_B serialized output frame will use the respective eTPU_A channel. |
| 12<br>ETPUA25 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_B serialized output frame will use the respective eTPU_A channel. |
| 13<br>ETPUA24 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_B serialized output frame will use the respective eTPU_A channel. |
| 14<br>ETPUA31 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_B serialized output frame will use the respective eTPU_A channel. |
| 15<br>ETPUA30 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_B serialized output frame will use the respective eTPU_A channel. |
| 16<br>ETPUA12 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_B serialized output frame will use the respective eTPU_A channel. |
| 17<br>ETPUA13 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_B serialized output frame will use the respective eTPU_A channel. |

Table continues on the next page...



**SIU\_ETPUAB field descriptions (continued)**

| Field         | Description                                                                                                                                                                                                                                                                         |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 18<br>ETPUA14 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_B serialized output frame will use the respective eTPU_A channel. |
| 19<br>ETPUA15 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_B serialized output frame will use the respective eTPU_A channel. |
| 20<br>ETPUA0  | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_B serialized output frame will use the respective eTPU_A channel. |
| 21<br>ETPUA1  | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_B serialized output frame will use the respective eTPU_A channel. |
| 22<br>ETPUA2  | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_B serialized output frame will use the respective eTPU_A channel. |
| 23<br>ETPUA3  | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_B serialized output frame will use the respective eTPU_A channel. |
| 24<br>ETPUA4  | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_B serialized output frame will use the respective eTPU_A channel. |
| 25<br>ETPUA5  | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_B serialized output frame will use the respective eTPU_A channel. |
| 26<br>ETPUA6  | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_B serialized output frame will use the respective eTPU_A channel. |
| 27<br>ETPUA7  | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_B serialized output frame will use the respective eTPU_A channel. |
| 28<br>ETPUA8  | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_B serialized output frame will use the respective eTPU_A channel. |
| 29<br>ETPUA9  | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_B serialized output frame will use the respective eTPU_A channel. |

*Table continues on the next page...*

**SIU\_ETPUAB field descriptions (continued)**

| Field         | Description                                                                                                                                                                                                                                                                             |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 30<br>ETPUA10 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_B output.<br><br>0 This bit in the DSPI_B serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_B serialized output frame will use the respective eTPU_A channel. |
| 31<br>ETPUA11 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_B output.<br><br>0 This bit in the DSPI_B serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_B serialized output frame will use the respective eTPU_A channel. |

**8.2.56 Serialized Output Signal Selection Register for DSPI\_B (SIU\_EMIOSB)**

Each bit in the DSPI serialized output frame can contain a signal routed from the output of one of three on-chip sources. The sources on this device are the eTPU module, eMIOS module, or a software-updated GPO data register. The mapping between module or data register and the output frame for a single DSPI channel is performed by independent 32-bit registers, one for each module plus one for the data register.

The following register is used by DSPI\_B to select the sources of the serialized output when running in DSI or CSI configuration.

Each register bit enables a path from the eTPU\_A channel, eMIOS channel and data register bit SIU\_DSPIBH/SIU\_DSPIBL to the equivalent bit position in the DSPI\_B serialized output frame. The user must ensure that bit selections from each of these registers do not overlap. Multiple sources are logically ORed, which provides the potential for combining outputs from multiple timer channels and data registers to produce more complex bit behavior.

Address: 0h base + D54h offset = D54h

| Bit   | 0         | 1         | 2         | 3         | 4         | 5         | 6         | 7         | 8         | 9         | 10        | 11         | 12        | 13        | 14        | 15        |
|-------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|------------|-----------|-----------|-----------|-----------|
| R     | EMIOS1_3_ | EMIOS1_2_ | EMIOS1_1_ | EMIOS1_0_ | EMIOS0_6_ | EMIOS0_5_ | EMIOS0_4_ | EMIOS0_3_ | EMIOS0_2_ | EMIOS0_1_ | EMIOS0_0_ | EMIOS0_23_ | EMIOS1_7_ | EMIOS1_6_ | EMIOS1_5_ | EMIOS1_4_ |
| W     | OUT0      | OUT1      | OUT2      | OUT3      | OUT4      | OUT5      | OUT6      | OUT7      | OUT8      | OUT9      | OUT10     | OUT11      | OUT12     | OUT13     | OUT14     | OUT15     |
| Reset | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0          | 0         | 0         | 0         | 0         |

| Bit   | 16         | 17        | 18        | 19        | 20        | 21        | 22        | 23        | 24        | 25        | 26        | 27        | 28        | 29        | 30        | 31        |
|-------|------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| R     | EMIOS0_23_ | EMIOS1_7_ | EMIOS1_6_ | EMIOS1_5_ | EMIOS1_4_ | EMIOS1_3_ | EMIOS1_2_ | EMIOS1_1_ | EMIOS1_0_ | EMIOS0_6_ | EMIOS0_5_ | EMIOS0_4_ | EMIOS0_3_ | EMIOS0_2_ | EMIOS0_1_ | EMIOS0_0_ |
| W     | OUT16      | OUT17     | OUT18     | OUT19     | OUT20     | OUT21     | OUT22     | OUT23     | OUT24     | OUT25     | OUT26     | OUT27     | OUT28     | OUT29     | OUT30     | OUT31     |
| Reset | 0          | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         |

### SIU\_EMIOSEB field descriptions

| Field          | Description                                                                                                                                                      |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>EMIOS1_3_ | eMIOS_1 Channel Select. Select an eMIOS_1 channel to source the serialized DSPI_B output.                                                                        |
| OUT0           | 0 This bit in the DSPI_B serialized output frame will not use eMIOS_1 channel 3.<br>1 This bit in the DSPI_B serialized output frame will use eMIOS_1 channel 3. |
| 1<br>EMIOS1_2_ | eMIOS_1 Channel Select. Select an eMIOS_1 channel to source the serialized DSPI_B output.                                                                        |
| OUT1           | 0 This bit in the DSPI_B serialized output frame will not use eMIOS_1 channel 2.<br>1 This bit in the DSPI_B serialized output frame will use eMIOS_1 channel 2. |
| 2<br>EMIOS1_1_ | eMIOS_1 Channel Select. Select an eMIOS_1 channel to source the serialized DSPI_B output.                                                                        |
| OUT2           | 0 This bit in the DSPI_B serialized output frame will not use eMIOS_1 channel 1.<br>1 This bit in the DSPI_B serialized output frame will use eMIOS_1 channel 1. |
| 3<br>EMIOS1_0_ | eMIOS_1 Channel Select. Select an eMIOS_1 channel to source the serialized DSPI_B output.                                                                        |
| OUT3           | 0 This bit in the DSPI_B serialized output frame will not use eMIOS_1 channel 0.<br>1 This bit in the DSPI_B serialized output frame will use eMIOS_1 channel 0. |
| 4<br>EMIOS0_6_ | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_B output.                                                                        |
| OUT4           | 0 This bit in the DSPI_B serialized output frame will not use eMIOS_0 channel 6.<br>1 This bit in the DSPI_B serialized output frame will use eMIOS_0 channel 6. |
| 5<br>EMIOS0_5_ | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_B output.                                                                        |
| OUT5           | 0 This bit in the DSPI_B serialized output frame will not use eMIOS_0 channel 5.<br>1 This bit in the DSPI_B serialized output frame will use eMIOS_0 channel 5. |
| 6<br>EMIOS0_4_ | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_B output.                                                                        |
| OUT6           | 0 This bit in the DSPI_B serialized output frame will not use eMIOS_0 channel 4.<br>1 This bit in the DSPI_B serialized output frame will use eMIOS_0 channel 4. |
| 7<br>EMIOS0_3_ | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_B output.                                                                        |
| OUT7           | 0 This bit in the DSPI_B serialized output frame will not use eMIOS_0 channel 3.<br>1 This bit in the DSPI_B serialized output frame will use eMIOS_0 channel 3. |
| 8<br>EMIOS0_2_ | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_B output.                                                                        |
| OUT8           | 0 This bit in the DSPI_B serialized output frame will not use eMIOS_0 channel 2.<br>1 This bit in the DSPI_B serialized output frame will use eMIOS_0 channel 2. |
| 9<br>EMIOS0_1_ | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_B output.                                                                        |
| OUT9           | 0 This bit in the DSPI_B serialized output frame will not use eMIOS_0 channel 1.<br>1 This bit in the DSPI_B serialized output frame will use eMIOS_0 channel 1. |

Table continues on the next page...

## SIU\_EMIOSB field descriptions (continued)

| Field                     | Description                                                                                                                                                                                                                                                     |
|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 10<br>EMIOS0_0_<br>OUT10  | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use eMIOS_0 channel 0.<br>1 This bit in the DSPI_B serialized output frame will use eMIOS_0 channel 0.   |
| 11<br>EMIOS0_23_<br>OUT11 | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use eMIOS_0 channel 23.<br>1 This bit in the DSPI_B serialized output frame will use eMIOS_0 channel 23. |
| 12<br>EMIOS1_7_<br>OUT12  | eMIOS_1 Channel Select. Select an eMIOS_1 channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use eMIOS_1 channel 7.<br>1 This bit in the DSPI_B serialized output frame will use eMIOS_1 channel 7.   |
| 13<br>EMIOS1_6_<br>OUT13  | eMIOS_1 Channel Select. Select an eMIOS_1 channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use eMIOS_1 channel 6.<br>1 This bit in the DSPI_B serialized output frame will use eMIOS_1 channel 6.   |
| 14<br>EMIOS1_5_<br>OUT14  | eMIOS_1 Channel Select. Select an eMIOS_1 channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use eMIOS_1 channel 5.<br>1 This bit in the DSPI_B serialized output frame will use eMIOS_1 channel 5.   |
| 15<br>EMIOS1_4_<br>OUT15  | eMIOS_1 Channel Select. Select an eMIOS_1 channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use eMIOS_1 channel 4.<br>1 This bit in the DSPI_B serialized output frame will use eMIOS_1 channel 4.   |
| 16<br>EMIOS0_23_<br>OUT16 | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use eMIOS_0 channel 23.<br>1 This bit in the DSPI_B serialized output frame will use eMIOS_0 channel 23. |
| 17<br>EMIOS1_7_<br>OUT17  | eMIOS_1 Channel Select. Select an eMIOS_1 channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use eMIOS_1 channel 7.<br>1 This bit in the DSPI_B serialized output frame will use eMIOS_1 channel 7.   |
| 18<br>EMIOS1_6_<br>OUT18  | eMIOS_1 Channel Select. Select an eMIOS_1 channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use eMIOS_1 channel 6.<br>1 This bit in the DSPI_B serialized output frame will use eMIOS_1 channel 6.   |
| 19<br>EMIOS1_5_<br>OUT19  | eMIOS_1 Channel Select. Select an eMIOS_1 channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use eMIOS_1 channel 5.<br>1 This bit in the DSPI_B serialized output frame will use eMIOS_1 channel 5.   |
| 20<br>EMIOS1_4_<br>OUT20  | eMIOS_1 Channel Select. Select an eMIOS_1 channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use eMIOS_1 channel 4.<br>1 This bit in the DSPI_B serialized output frame will use eMIOS_1 channel 4.   |
| 21<br>EMIOS1_3_<br>OUT21  | eMIOS_1 Channel Select. Select an eMIOS_1 channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use eMIOS_1 channel 3.<br>1 This bit in the DSPI_B serialized output frame will use eMIOS_1 channel 3.   |

Table continues on the next page...

## SIU\_EMIOSB field descriptions (continued)

| Field                    | Description                                                                                                                                                                                                                                                   |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 22<br>EMIOS1_2_<br>OUT22 | eMIOS_1 Channel Select. Select an eMIOS_1 channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use eMIOS_1 channel 2.<br>1 This bit in the DSPI_B serialized output frame will use eMIOS_1 channel 2. |
| 23<br>EMIOS1_1_<br>OUT23 | eMIOS_1 Channel Select. Select an eMIOS_1 channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use eMIOS_1 channel 1.<br>1 This bit in the DSPI_B serialized output frame will use eMIOS_1 channel 1. |
| 24<br>EMIOS1_0_<br>OUT24 | eMIOS_1 Channel Select. Select an eMIOS_1 channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use eMIOS_1 channel 0.<br>1 This bit in the DSPI_B serialized output frame will use eMIOS_1 channel 0. |
| 25<br>EMIOS0_6_<br>OUT25 | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use eMIOS_0 channel 6.<br>1 This bit in the DSPI_B serialized output frame will use eMIOS_0 channel 6. |
| 26<br>EMIOS0_5_<br>OUT26 | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use eMIOS_0 channel 5.<br>1 This bit in the DSPI_B serialized output frame will use eMIOS_0 channel 5. |
| 27<br>EMIOS0_4_<br>OUT27 | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use eMIOS_0 channel 4.<br>1 This bit in the DSPI_B serialized output frame will use eMIOS_0 channel 4. |
| 28<br>EMIOS0_3_<br>OUT28 | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use eMIOS_0 channel 3.<br>1 This bit in the DSPI_B serialized output frame will use eMIOS_0 channel 3. |
| 29<br>EMIOS0_2_<br>OUT29 | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use eMIOS_0 channel 2.<br>1 This bit in the DSPI_B serialized output frame will use eMIOS_0 channel 2. |
| 30<br>EMIOS0_1_<br>OUT30 | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use eMIOS_0 channel 1.<br>1 This bit in the DSPI_B serialized output frame will use eMIOS_0 channel 1. |
| 31<br>EMIOS0_0_<br>OUT31 | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_B output.<br>0 This bit in the DSPI_B serialized output frame will not use eMIOS_0 channel 0.<br>1 This bit in the DSPI_B serialized output frame will use eMIOS_0 channel 0. |

## 8.2.57 Serialized Output Signal Selection Register for DSPI\_B (SIU\_DSPIBHLB)

Each bit in the DSPI serialized output frame can contain a signal routed from the output of one of three on-chip sources. The sources on this device are the eTPU module, eMIOS module, or a software-updated GPO data register. The mapping between module or data register and the output frame for a single DSPI channel is performed by independent 32-bit registers, one for each module plus one for the data register.

The following register is used by DSPI\_B to select the sources of the serialized output when running in DSI or CSI configuration.

Each register bit enables a path from the eTPU\_A channel, eMIOS channel and data register bit SIU\_DSPIBH/SIU\_DSPIBL to the equivalent bit position in the DSPI\_B serialized output frame. The user must ensure that bit selections from each of these registers do not overlap. Multiple sources are logically ORed, which provides the potential for combining outputs from multiple timer channels and data registers to produce more complex bit behavior.

Address: 0h base + D58h offset = D58h

|       |         |         |         |         |         |         |         |         |         |         |          |          |          |          |          |          |
|-------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|----------|----------|----------|----------|----------|----------|
| Bit   | 0       | 1       | 2       | 3       | 4       | 5       | 6       | 7       | 8       | 9       | 10       | 11       | 12       | 13       | 14       | 15       |
| R     | DSPIBH0 | DSPIBH1 | DSPIBH2 | DSPIBH3 | DSPIBH4 | DSPIBH5 | DSPIBH6 | DSPIBH7 | DSPIBH8 | DSPIBH9 | DSPIBH10 | DSPIBH11 | DSPIBH12 | DSPIBH13 | DSPIBH14 | DSPIBH15 |
| W     | DSPIBH0 | DSPIBH1 | DSPIBH2 | DSPIBH3 | DSPIBH4 | DSPIBH5 | DSPIBH6 | DSPIBH7 | DSPIBH8 | DSPIBH9 | DSPIBH10 | DSPIBH11 | DSPIBH12 | DSPIBH13 | DSPIBH14 | DSPIBH15 |
| Reset | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0        | 0        | 0        | 0        | 0        | 0        |

|       |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |
|-------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| Bit   | 16       | 17       | 18       | 19       | 20       | 21       | 22       | 23       | 24       | 25       | 26       | 27       | 28       | 29       | 30       | 31       |
| R     | DSPIBL16 | DSPIBL17 | DSPIBL18 | DSPIBL19 | DSPIBL20 | DSPIBL21 | DSPIBL22 | DSPIBL23 | DSPIBL24 | DSPIBL25 | DSPIBL26 | DSPIBL27 | DSPIBL28 | DSPIBL29 | DSPIBL30 | DSPIBL31 |
| W     | DSPIBL16 | DSPIBL17 | DSPIBL18 | DSPIBL19 | DSPIBL20 | DSPIBL21 | DSPIBL22 | DSPIBL23 | DSPIBL24 | DSPIBL25 | DSPIBL26 | DSPIBL27 | DSPIBL28 | DSPIBL29 | DSPIBL30 | DSPIBL31 |
| Reset | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        |

### SIU\_DSPIBHLB field descriptions

| Field        | Description                                                                                                                                                                                                                                                                                                           |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>DSPIBH0 | SIU_DSPIBH Select. Select one bit of the DSPI_B GPO Mask Output High Register to source the serialized DSPI_B output.<br><br>0 This bit in the DSPI_B serialized output frame will not use the respective SIU_DSPIBH bit.<br>1 This bit in the DSPI_B serialized output frame will use the respective SIU_DSPIBH bit. |
| 1<br>DSPIBH1 | SIU_DSPIBH Select. Select one bit of the DSPI_B GPO Mask Output High Register to source the serialized DSPI_B output.                                                                                                                                                                                                 |

Table continues on the next page...

**SIU\_DSPIBHLB field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                                                                           |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | 0 This bit in the DSPI_B serialized output frame will not use the respective SIU_DSPIBH bit.<br>1 This bit in the DSPI_B serialized output frame will use the respective SIU_DSPIBH bit.                                                                                                                              |
| 2<br>DSPIBH2   | SIU_DSPIBH Select. Select one bit of the DSPI_B GPO Mask Output High Register to source the serialized DSPI_B output.<br><br>0 This bit in the DSPI_B serialized output frame will not use the respective SIU_DSPIBH bit.<br>1 This bit in the DSPI_B serialized output frame will use the respective SIU_DSPIBH bit. |
| 3<br>DSPIBH3   | SIU_DSPIBH Select. Select one bit of the DSPI_B GPO Mask Output High Register to source the serialized DSPI_B output.<br><br>0 This bit in the DSPI_B serialized output frame will not use the respective SIU_DSPIBH bit.<br>1 This bit in the DSPI_B serialized output frame will use the respective SIU_DSPIBH bit. |
| 4<br>DSPIBH4   | SIU_DSPIBH Select. Select one bit of the DSPI_B GPO Mask Output High Register to source the serialized DSPI_B output.<br><br>0 This bit in the DSPI_B serialized output frame will not use the respective SIU_DSPIBH bit.<br>1 This bit in the DSPI_B serialized output frame will use the respective SIU_DSPIBH bit. |
| 5<br>DSPIBH5   | SIU_DSPIBH Select. Select one bit of the DSPI_B GPO Mask Output High Register to source the serialized DSPI_B output.<br><br>0 This bit in the DSPI_B serialized output frame will not use the respective SIU_DSPIBH bit.<br>1 This bit in the DSPI_B serialized output frame will use the respective SIU_DSPIBH bit. |
| 6<br>DSPIBH6   | SIU_DSPIBH Select. Select one bit of the DSPI_B GPO Mask Output High Register to source the serialized DSPI_B output.<br><br>0 This bit in the DSPI_B serialized output frame will not use the respective SIU_DSPIBH bit.<br>1 This bit in the DSPI_B serialized output frame will use the respective SIU_DSPIBH bit. |
| 7<br>DSPIBH7   | SIU_DSPIBH Select. Select one bit of the DSPI_B GPO Mask Output High Register to source the serialized DSPI_B output.<br><br>0 This bit in the DSPI_B serialized output frame will not use the respective SIU_DSPIBH bit.<br>1 This bit in the DSPI_B serialized output frame will use the respective SIU_DSPIBH bit. |
| 8<br>DSPIBH8   | SIU_DSPIBH Select. Select one bit of the DSPI_B GPO Mask Output High Register to source the serialized DSPI_B output.<br><br>0 This bit in the DSPI_B serialized output frame will not use the respective SIU_DSPIBH bit.<br>1 This bit in the DSPI_B serialized output frame will use the respective SIU_DSPIBH bit. |
| 9<br>DSPIBH9   | SIU_DSPIBH Select. Select one bit of the DSPI_B GPO Mask Output High Register to source the serialized DSPI_B output.<br><br>0 This bit in the DSPI_B serialized output frame will not use the respective SIU_DSPIBH bit.<br>1 This bit in the DSPI_B serialized output frame will use the respective SIU_DSPIBH bit. |
| 10<br>DSPIBH10 | SIU_DSPIBH Select. Select one bit of the DSPI_B GPO Mask Output High Register to source the serialized DSPI_B output.<br><br>0 This bit in the DSPI_B serialized output frame will not use the respective SIU_DSPIBH bit.<br>1 This bit in the DSPI_B serialized output frame will use the respective SIU_DSPIBH bit. |
| 11<br>DSPIBH11 | SIU_DSPIBH Select. Select one bit of the DSPI_B GPO Mask Output High Register to source the serialized DSPI_B output.                                                                                                                                                                                                 |

*Table continues on the next page...*

**SIU\_DSPIBHLB field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                                                                           |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | 0 This bit in the DSPI_B serialized output frame will not use the respective SIU_DSPIBH bit.<br>1 This bit in the DSPI_B serialized output frame will use the respective SIU_DSPIBH bit.                                                                                                                              |
| 12<br>DSPIBH12 | SIU_DSPIBH Select. Select one bit of the DSPI_B GPO Mask Output High Register to source the serialized DSPI_B output.<br><br>0 This bit in the DSPI_B serialized output frame will not use the respective SIU_DSPIBH bit.<br>1 This bit in the DSPI_B serialized output frame will use the respective SIU_DSPIBH bit. |
| 13<br>DSPIBH13 | SIU_DSPIBH Select. Select one bit of the DSPI_B GPO Mask Output High Register to source the serialized DSPI_B output.<br><br>0 This bit in the DSPI_B serialized output frame will not use the respective SIU_DSPIBH bit.<br>1 This bit in the DSPI_B serialized output frame will use the respective SIU_DSPIBH bit. |
| 14<br>DSPIBH14 | SIU_DSPIBH Select. Select one bit of the DSPI_B GPO Mask Output High Register to source the serialized DSPI_B output.<br><br>0 This bit in the DSPI_B serialized output frame will not use the respective SIU_DSPIBH bit.<br>1 This bit in the DSPI_B serialized output frame will use the respective SIU_DSPIBH bit. |
| 15<br>DSPIBH15 | SIU_DSPIBH Select. Select one bit of the DSPI_B GPO Mask Output High Register to source the serialized DSPI_B output.<br><br>0 This bit in the DSPI_B serialized output frame will not use the respective SIU_DSPIBH bit.<br>1 This bit in the DSPI_B serialized output frame will use the respective SIU_DSPIBH bit. |
| 16<br>DSPIBL16 | SIU_DSPIBL Select. Select one bit of the DSPI_B GPO Mask Output Low Register to source the serialized DSPI_B output.<br><br>0 This bit in the DSPI_B serialized output frame will not use the respective SIU_DSPIBL bit.<br>1 This bit in the DSPI_B serialized output frame will use the respective SIU_DSPIBL bit.  |
| 17<br>DSPIBL17 | SIU_DSPIBL Select. Select one bit of the DSPI_B GPO Mask Output Low Register to source the serialized DSPI_B output.<br><br>0 This bit in the DSPI_B serialized output frame will not use the respective SIU_DSPIBL bit.<br>1 This bit in the DSPI_B serialized output frame will use the respective SIU_DSPIBL bit.  |
| 18<br>DSPIBL18 | SIU_DSPIBL Select. Select one bit of the DSPI_B GPO Mask Output Low Register to source the serialized DSPI_B output.<br><br>0 This bit in the DSPI_B serialized output frame will not use the respective SIU_DSPIBL bit.<br>1 This bit in the DSPI_B serialized output frame will use the respective SIU_DSPIBL bit.  |
| 19<br>DSPIBL19 | SIU_DSPIBL Select. Select one bit of the DSPI_B GPO Mask Output Low Register to source the serialized DSPI_B output.<br><br>0 This bit in the DSPI_B serialized output frame will not use the respective SIU_DSPIBL bit.<br>1 This bit in the DSPI_B serialized output frame will use the respective SIU_DSPIBL bit.  |
| 20<br>DSPIBL20 | SIU_DSPIBL Select. Select one bit of the DSPI_B GPO Mask Output Low Register to source the serialized DSPI_B output.<br><br>0 This bit in the DSPI_B serialized output frame will not use the respective SIU_DSPIBL bit.<br>1 This bit in the DSPI_B serialized output frame will use the respective SIU_DSPIBL bit.  |
| 21<br>DSPIBL21 | SIU_DSPIBL Select. Select one bit of the DSPI_B GPO Mask Output Low Register to source the serialized DSPI_B output.                                                                                                                                                                                                  |

*Table continues on the next page...*



**SIU\_DSPIBLB field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                                                                          |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | 0 This bit in the DSPI_B serialized output frame will not use the respective SIU_DSPIBL bit.<br>1 This bit in the DSPI_B serialized output frame will use the respective SIU_DSPIBL bit.                                                                                                                             |
| 22<br>DSPIBL22 | SIU_DSPIBL Select. Select one bit of the DSPI_B GPO Mask Output Low Register to source the serialized DSPI_B output.<br><br>0 This bit in the DSPI_B serialized output frame will not use the respective SIU_DSPIBL bit.<br>1 This bit in the DSPI_B serialized output frame will use the respective SIU_DSPIBL bit. |
| 23<br>DSPIBL23 | SIU_DSPIBL Select. Select one bit of the DSPI_B GPO Mask Output Low Register to source the serialized DSPI_B output.<br><br>0 This bit in the DSPI_B serialized output frame will not use the respective SIU_DSPIBL bit.<br>1 This bit in the DSPI_B serialized output frame will use the respective SIU_DSPIBL bit. |
| 24<br>DSPIBL24 | SIU_DSPIBL Select. Select one bit of the DSPI_B GPO Mask Output Low Register to source the serialized DSPI_B output.<br><br>0 This bit in the DSPI_B serialized output frame will not use the respective SIU_DSPIBL bit.<br>1 This bit in the DSPI_B serialized output frame will use the respective SIU_DSPIBL bit. |
| 25<br>DSPIBL25 | SIU_DSPIBL Select. Select one bit of the DSPI_B GPO Mask Output Low Register to source the serialized DSPI_B output.<br><br>0 This bit in the DSPI_B serialized output frame will not use the respective SIU_DSPIBL bit.<br>1 This bit in the DSPI_B serialized output frame will use the respective SIU_DSPIBL bit. |
| 26<br>DSPIBL26 | SIU_DSPIBL Select. Select one bit of the DSPI_B GPO Mask Output Low Register to source the serialized DSPI_B output.<br><br>0 This bit in the DSPI_B serialized output frame will not use the respective SIU_DSPIBL bit.<br>1 This bit in the DSPI_B serialized output frame will use the respective SIU_DSPIBL bit. |
| 27<br>DSPIBL27 | SIU_DSPIBL Select. Select one bit of the DSPI_B GPO Mask Output Low Register to source the serialized DSPI_B output.<br><br>0 This bit in the DSPI_B serialized output frame will not use the respective SIU_DSPIBL bit.<br>1 This bit in the DSPI_B serialized output frame will use the respective SIU_DSPIBL bit. |
| 28<br>DSPIBL28 | SIU_DSPIBL Select. Select one bit of the DSPI_B GPO Mask Output Low Register to source the serialized DSPI_B output.<br><br>0 This bit in the DSPI_B serialized output frame will not use the respective SIU_DSPIBL bit.<br>1 This bit in the DSPI_B serialized output frame will use the respective SIU_DSPIBL bit. |
| 29<br>DSPIBL29 | SIU_DSPIBL Select. Select one bit of the DSPI_B GPO Mask Output Low Register to source the serialized DSPI_B output.<br><br>0 This bit in the DSPI_B serialized output frame will not use the respective SIU_DSPIBL bit.<br>1 This bit in the DSPI_B serialized output frame will use the respective SIU_DSPIBL bit. |
| 30<br>DSPIBL30 | SIU_DSPIBL Select. Select one bit of the DSPI_B GPO Mask Output Low Register to source the serialized DSPI_B output.<br><br>0 This bit in the DSPI_B serialized output frame will not use the respective SIU_DSPIBL bit.<br>1 This bit in the DSPI_B serialized output frame will use the respective SIU_DSPIBL bit. |
| 31<br>DSPIBL31 | SIU_DSPIBL Select. Select one bit of the DSPI_B GPO Mask Output Low Register to source the serialized DSPI_B output.                                                                                                                                                                                                 |

*Table continues on the next page...*

**SIU\_DSPIBHLB field descriptions (continued)**

| Field | Description                                                                                |
|-------|--------------------------------------------------------------------------------------------|
| 0     | This bit in the DSPI_B serialized output frame will not use the respective SIU_DSPIBL bit. |
| 1     | This bit in the DSPI_B serialized output frame will use the respective SIU_DSPIBL bit.     |

**8.2.58 Serialized Output Signal Selection Register for DSPI\_C (SIU\_ETPUAC)**

Each bit in the DSPI serialized output frame can contain a signal routed from the output of one of three on-chip sources. The sources on this device are the eTPU module, eMIOS module, or a software-updated GPO data register. The mapping between module or data register and the output frame for a single DSPI channel is performed by independent 32-bit registers, one for each module plus one for the data register.

The following register is used by DSPI\_C to select the sources of the serialized output when running in DSI or CSI configuration.

Each register bit enables a path from the eTPU\_A/B channel, eMIOS channel and data register bit SIU\_DSPICH/SIU\_DSPICL to the equivalent bit position in the DSPI\_C serialized output frame. The user must ensure that bit selections from each of these registers do not overlap. Multiple sources are logically ORed, which provides the potential for combining outputs from multiple timer channels and data registers to produce more complex bit behavior.

Address: 0h base + D60h offset = D60h

|       |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
|-------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| Bit   | 0       | 1       | 2       | 3       | 4       | 5       | 6       | 7       | 8       | 9       | 10      | 11      | 12      | 13      | 14      | 15      |
| R     | ETPUA12 | ETPUA13 | ETPUA14 | ETPUA15 | ETPUA0  | ETPUA1  | ETPUA2  | ETPUA3  | ETPUA4  | ETPUA5  | ETPUA6  | ETPUA7  | ETPUA8  | ETPUA9  | ETPUA10 | ETPUA11 |
| W     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| Reset | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       |
| Bit   | 16      | 17      | 18      | 19      | 20      | 21      | 22      | 23      | 24      | 25      | 26      | 27      | 28      | 29      | 30      | 31      |
| R     | ETPUA23 | ETPUA22 | ETPUA21 | ETPUA20 | ETPUA19 | ETPUA18 | ETPUA17 | ETPUA16 | ETPUA29 | ETPUA28 | ETPUA27 | ETPUA26 | ETPUA25 | ETPUA24 | ETPUA31 | ETPUA30 |
| W     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| Reset | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       |

**SIU\_ETPUAC field descriptions**

| Field        | Description                                                                             |
|--------------|-----------------------------------------------------------------------------------------|
| 0<br>ETPUA12 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_C output. |

*Table continues on the next page...*

**SIU\_ETPUAC field descriptions (continued)**

| Field        | Description                                                                                                                                                                                                                                                                             |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|              | 0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_A channel.                                                                                                |
| 1<br>ETPUA13 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_A channel. |
| 2<br>ETPUA14 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_A channel. |
| 3<br>ETPUA15 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_A channel. |
| 4<br>ETPUA0  | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_A channel. |
| 5<br>ETPUA1  | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_A channel. |
| 6<br>ETPUA2  | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_A channel. |
| 7<br>ETPUA3  | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_A channel. |
| 8<br>ETPUA4  | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_A channel. |
| 9<br>ETPUA5  | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_A channel. |
| 10<br>ETPUA6 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_A channel. |
| 11<br>ETPUA7 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_A channel. |
| 12<br>ETPUA8 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_C output.                                                                                                                                                                                                 |

*Table continues on the next page...*

## SIU\_ETPUAC field descriptions (continued)

| Field         | Description                                                                                                                                                                                                                                                                             |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               | 0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_A channel.                                                                                                |
| 13<br>ETPUA9  | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_A channel. |
| 14<br>ETPUA10 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_A channel. |
| 15<br>ETPUA11 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_A channel. |
| 16<br>ETPUA23 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_A channel. |
| 17<br>ETPUA22 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_A channel. |
| 18<br>ETPUA21 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_A channel. |
| 19<br>ETPUA20 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_A channel. |
| 20<br>ETPUA19 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_A channel. |
| 21<br>ETPUA18 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_A channel. |
| 22<br>ETPUA17 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_A channel. |
| 23<br>ETPUA16 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_A channel. |
| 24<br>ETPUA29 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_C output.                                                                                                                                                                                                 |

Table continues on the next page...

**SIU\_ETPUAC field descriptions (continued)**

| Field         | Description                                                                                                                                                                                                                                                                         |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               | 0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_A channel.                                                                                            |
| 25<br>ETPUA28 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_A channel. |
| 26<br>ETPUA27 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_A channel. |
| 27<br>ETPUA26 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_A channel. |
| 28<br>ETPUA25 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_A channel. |
| 29<br>ETPUA24 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_A channel. |
| 30<br>ETPUA31 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_A channel. |
| 31<br>ETPUA30 | eTPU_A Channel Select. Select an eTPU_A channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_A channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_A channel. |

**8.2.59 Serialized Output Signal Selection Register for DSPI\_C (SIU\_EMIOSC)**

Each bit in the DSPI serialized output frame can contain a signal routed from the output of one of three on-chip sources. The sources on this device are the eTPU module, eMIOS module, or a software-updated GPO data register. The mapping between module or data register and the output frame for a single DSPI channel is performed by independent 32-bit registers, one for each module plus one for the data register.

The following register is used by DSPI\_C to select the sources of the serialized output when running in DSI or CSI configuration.

## Memory Map and Register Definition

Each register bit enables a path from the eTPU\_A/B channel, eMIOS channel and data register bit SIU\_DSPICH/SIU\_DSPICL to the equivalent bit position in the DSPI\_C serialized output frame. The user must ensure that bit selections from each of these registers do not overlap. Multiple sources are logically ORed, which provides the potential for combining outputs from multiple timer channels and data registers to produce more complex bit behavior.

Address: 0h base + D64h offset = D64h

| Bit   | 0         | 1         | 2         | 3         | 4          | 5         | 6         | 7         | 8         | 9         | 10        | 11        | 12        | 13        | 14        | 15        |
|-------|-----------|-----------|-----------|-----------|------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| R     |           |           |           |           |            |           |           |           |           |           |           |           |           |           |           |           |
| W     | EMIOS1_4_ | EMIOS1_5_ | EMIOS1_6_ | EMIOS1_7_ | EMIOS0_23_ | EMIOS0_0_ | EMIOS0_1_ | EMIOS0_2_ | EMIOS0_3_ | EMIOS0_4_ | EMIOS0_5_ | EMIOS0_6_ | EMIOS1_0_ | EMIOS1_1_ | EMIOS1_2_ | EMIOS1_3_ |
|       | OUT0      | OUT1      | OUT2      | OUT3      | OUT4       | OUT5      | OUT6      | OUT7      | OUT8      | OUT9      | OUT10     | OUT11     | OUT12     | OUT13     | OUT14     | OUT15     |
| Reset | 0         | 0         | 0         | 0         | 0          | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         |

| Bit   | 16        | 17        | 18        | 19        | 20        | 21        | 22        | 23        | 24        | 25        | 26        | 27        | 28        | 29        | 30        | 31         |
|-------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|------------|
| R     |           |           |           |           |           |           |           |           |           |           |           |           |           |           |           |            |
| W     | EMIOS0_0_ | EMIOS0_1_ | EMIOS0_2_ | EMIOS0_3_ | EMIOS0_4_ | EMIOS0_5_ | EMIOS0_6_ | EMIOS1_0_ | EMIOS1_1_ | EMIOS1_2_ | EMIOS1_3_ | EMIOS1_4_ | EMIOS1_5_ | EMIOS1_6_ | EMIOS1_7_ | EMIOS0_23_ |
|       | OUT16     | OUT17     | OUT18     | OUT19     | OUT20     | OUT21     | OUT22     | OUT23     | OUT24     | OUT25     | OUT26     | OUT27     | OUT28     | OUT29     | OUT30     | OUT31      |
| Reset | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0         | 0          |

### SIU\_EMIOSC field descriptions

| Field           | Description                                                                                                                                                        |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>EMIOS1_4_  | eMIOS_1 Channel Select. Select an eMIOS_1 channel to source the serialized DSPI_C output.                                                                          |
| OUT0            | 0 This bit in the DSPI_C serialized output frame will not use eMIOS_1 channel 4.<br>1 This bit in the DSPI_C serialized output frame will use eMIOS_1 channel 4.   |
| 1<br>EMIOS1_5_  | eMIOS_1 Channel Select. Select an eMIOS_1 channel to source the serialized DSPI_C output.                                                                          |
| OUT1            | 0 This bit in the DSPI_C serialized output frame will not use eMIOS_1 channel 5.<br>1 This bit in the DSPI_C serialized output frame will use eMIOS_1 channel .5   |
| 2<br>EMIOS1_6_  | eMIOS_1 Channel Select. Select an eMIOS_1 channel to source the serialized DSPI_C output.                                                                          |
| OUT2            | 0 This bit in the DSPI_C serialized output frame will not use eMIOS_1 channel 6.<br>1 This bit in the DSPI_C serialized output frame will use eMIOS_1 channel 6.   |
| 3<br>EMIOS1_7_  | eMIOS_1 Channel Select. Select an eMIOS_1 channel to source the serialized DSPI_C output.                                                                          |
| OUT3            | 0 This bit in the DSPI_C serialized output frame will not use eMIOS_1 channel 7.<br>1 This bit in the DSPI_C serialized output frame will use eMIOS_1 channel 7.   |
| 4<br>EMIOS0_23_ | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_C output.                                                                          |
| OUT4            | 0 This bit in the DSPI_C serialized output frame will not use eMIOS_0 channel 23.<br>1 This bit in the DSPI_C serialized output frame will use eMIOS_0 channel 23. |

Table continues on the next page...

**SIU\_EMIOSC field descriptions (continued)**

| Field                | Description                                                                                                                                                                                                                                                   |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 5<br>EMIOS0_0_OUT5   | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use eMIOS_0 channel 0.<br>1 This bit in the DSPI_C serialized output frame will use eMIOS_0 channel 0. |
| 6<br>EMIOS0_1_OUT6   | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use eMIOS_0 channel 1.<br>1 This bit in the DSPI_C serialized output frame will use eMIOS_0 channel 1. |
| 7<br>EMIOS0_2_OUT7   | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use eMIOS_0 channel 2.<br>1 This bit in the DSPI_C serialized output frame will use eMIOS_0 channel 2. |
| 8<br>EMIOS0_3_OUT8   | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use eMIOS_0 channel 3.<br>1 This bit in the DSPI_C serialized output frame will use eMIOS_0 channel 3. |
| 9<br>EMIOS0_4_OUT9   | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use eMIOS_0 channel 4.<br>1 This bit in the DSPI_C serialized output frame will use eMIOS_0 channel 4. |
| 10<br>EMIOS0_5_OUT10 | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use eMIOS_0 channel 5.<br>1 This bit in the DSPI_C serialized output frame will use eMIOS_0 channel 5. |
| 11<br>EMIOS0_6_OUT11 | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use eMIOS_0 channel 6.<br>1 This bit in the DSPI_C serialized output frame will use eMIOS_0 channel 6. |
| 12<br>EMIOS1_0_OUT12 | eMIOS_1 Channel Select. Select an eMIOS_1 channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use eMIOS_1 channel 0.<br>1 This bit in the DSPI_C serialized output frame will use eMIOS_1 channel 0. |
| 13<br>EMIOS1_1_OUT13 | eMIOS_1 Channel Select. Select an eMIOS_1 channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use eMIOS_1 channel 1.<br>1 This bit in the DSPI_C serialized output frame will use eMIOS_1 channel 1. |
| 14<br>EMIOS1_2_OUT14 | eMIOS_1 Channel Select. Select an eMIOS_1 channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use eMIOS_1 channel 2.<br>1 This bit in the DSPI_C serialized output frame will use eMIOS_1 channel 2. |
| 15<br>EMIOS1_3_OUT15 | eMIOS_1 Channel Select. Select an eMIOS_1 channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use eMIOS_1 channel 3.<br>1 This bit in the DSPI_C serialized output frame will use eMIOS_1 channel 3. |
| 16<br>EMIOS0_0_OUT16 | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use eMIOS_0 channel 0.<br>1 This bit in the DSPI_C serialized output frame will use eMIOS_0 channel 0. |

*Table continues on the next page...*

## SIU\_EMIOSC field descriptions (continued)

| Field                | Description                                                                                                                                                                                                                                                   |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 17<br>EMIOS0_1_OUT17 | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use eMIOS_0 channel 1.<br>1 This bit in the DSPI_C serialized output frame will use eMIOS_0 channel 1. |
| 18<br>EMIOS0_2_OUT18 | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use eMIOS_0 channel 2.<br>1 This bit in the DSPI_C serialized output frame will use eMIOS_0 channel 2. |
| 19<br>EMIOS0_3_OUT19 | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use eMIOS_0 channel 3.<br>1 This bit in the DSPI_C serialized output frame will use eMIOS_0 channel 3. |
| 20<br>EMIOS0_4_OUT20 | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use eMIOS_0 channel 4.<br>1 This bit in the DSPI_C serialized output frame will use eMIOS_0 channel 4. |
| 21<br>EMIOS0_5_OUT21 | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use eMIOS_0 channel 5.<br>1 This bit in the DSPI_C serialized output frame will use eMIOS_0 channel 5. |
| 22<br>EMIOS0_6_OUT22 | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use eMIOS_0 channel 6.<br>1 This bit in the DSPI_C serialized output frame will use eMIOS_0 channel 6. |
| 23<br>EMIOS1_0_OUT23 | eMIOS_1 Channel Select. Select an eMIOS_1 channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use eMIOS_1 channel 0.<br>1 This bit in the DSPI_C serialized output frame will use eMIOS_1 channel 0. |
| 24<br>EMIOS1_1_OUT24 | eMIOS_1 Channel Select. Select an eMIOS_1 channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use eMIOS_1 channel 1.<br>1 This bit in the DSPI_C serialized output frame will use eMIOS_1 channel 1. |
| 25<br>EMIOS1_2_OUT25 | eMIOS_1 Channel Select. Select an eMIOS_1 channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use eMIOS_1 channel 2.<br>1 This bit in the DSPI_C serialized output frame will use eMIOS_1 channel 2. |
| 26<br>EMIOS1_3_OUT26 | eMIOS_1 Channel Select. Select an eMIOS_1 channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use eMIOS_1 channel 3.<br>1 This bit in the DSPI_C serialized output frame will use eMIOS_1 channel 3. |
| 27<br>EMIOS1_4_OUT27 | eMIOS_1 Channel Select. Select an eMIOS_1 channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use eMIOS_1 channel 4.<br>1 This bit in the DSPI_C serialized output frame will use eMIOS_1 channel 4. |
| 28<br>EMIOS1_5_OUT28 | eMIOS_1 Channel Select. Select an eMIOS_1 channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use eMIOS_1 channel 5.<br>1 This bit in the DSPI_C serialized output frame will use eMIOS_1 channel 5. |

Table continues on the next page...



**SIU\_EMIOSC field descriptions (continued)**

| Field                 | Description                                                                                                                                                                                                                                                     |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 29<br>EMIOS1_6_OUT29  | eMIOS_1 Channel Select. Select an eMIOS_1 channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use eMIOS_1 channel 6.<br>1 This bit in the DSPI_C serialized output frame will use eMIOS_1 channel 6.   |
| 30<br>EMIOS1_7_OUT30  | eMIOS_1 Channel Select. Select an eMIOS_1 channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use eMIOS_1 channel 7.<br>1 This bit in the DSPI_C serialized output frame will use eMIOS_1 channel 7.   |
| 31<br>EMIOS0_23_OUT31 | eMIOS_0 Channel Select. Select an eMIOS_0 channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use eMIOS_0 channel 23.<br>1 This bit in the DSPI_C serialized output frame will use eMIOS_0 channel 23. |

**8.2.60 Serialized Output Signal Selection Register for DSPI\_C (SIU\_DSPICHL)**

Each bit in the DSPI serialized output frame can contain a signal routed from the output of one of three on-chip sources. The sources on this device are the eTPU module, eMIOS module, or a software-updated GPO data register. The mapping between module or data register and the output frame for a single DSPI channel is performed by independent 32-bit registers, one for each module plus one for the data register.

The following register is used by DSPI\_C to select the sources of the serialized output when running in DSI or CSI configuration.

Each register bit enables a path from the eTPU\_A/B channel, eMIOS channel and data register bit SIU\_DSPICH/SIU\_DSPICL to the equivalent bit position in the DSPI\_C serialized output frame. The user must ensure that bit selections from each of these registers do not overlap. Multiple sources are logically ORed, which provides the potential for combining outputs from multiple timer channels and data registers to produce more complex bit behavior.

Address: 0h base + D68h offset = D68h

| Bit   | 0       | 1       | 2       | 3       | 4       | 5       | 6       | 7       | 8       | 9       | 10       | 11       | 12       | 13       | 14       | 15       |
|-------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|----------|----------|----------|----------|----------|----------|
| R     | DSPICH0 | DSPICH1 | DSPICH2 | DSPICH3 | DSPICH4 | DSPICH5 | DSPICH6 | DSPICH7 | DSPICH8 | DSPICH9 | DSPICH10 | DSPICH11 | DSPICH12 | DSPICH13 | DSPICH14 | DSPICH15 |
| W     |         |         |         |         |         |         |         |         |         |         |          |          |          |          |          |          |
| Reset | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0        | 0        | 0        | 0        | 0        | 0        |

## Memory Map and Register Definition

|       |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |
|-------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| Bit   | 16       | 17       | 18       | 19       | 20       | 21       | 22       | 23       | 24       | 25       | 26       | 27       | 28       | 29       | 30       | 31       |
| R     | DSPICL16 | DSPICL17 | DSPICL18 | DSPICL19 | DSPICL20 | DSPICL21 | DSPICL22 | DSPICL23 | DSPICL24 | DSPICL25 | DSPICL26 | DSPICL27 | DSPICL28 | DSPICL29 | DSPICL30 | DSPICL31 |
| W     |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |
| Reset | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        |

### SIU\_DSPICHL field descriptions

| Field         | Description                                                                                                                                                                                                                                                                                                              |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>DSPICHL0 | SIU_DSPICHL Select. Select one bit of the DSPI_C GPO Mask Output High Register to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective SIU_DSPICHL bit.<br>1 This bit in the DSPI_C serialized output frame will use the respective SIU_DSPICHL bit. |
| 1<br>DSPICHL1 | SIU_DSPICHL Select. Select one bit of the DSPI_C GPO Mask Output High Register to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective SIU_DSPICHL bit.<br>1 This bit in the DSPI_C serialized output frame will use the respective SIU_DSPICHL bit. |
| 2<br>DSPICHL2 | SIU_DSPICHL Select. Select one bit of the DSPI_C GPO Mask Output High Register to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective SIU_DSPICHL bit.<br>1 This bit in the DSPI_C serialized output frame will use the respective SIU_DSPICHL bit. |
| 3<br>DSPICHL3 | SIU_DSPICHL Select. Select one bit of the DSPI_C GPO Mask Output High Register to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective SIU_DSPICHL bit.<br>1 This bit in the DSPI_C serialized output frame will use the respective SIU_DSPICHL bit. |
| 4<br>DSPICHL4 | SIU_DSPICHL Select. Select one bit of the DSPI_C GPO Mask Output High Register to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective SIU_DSPICHL bit.<br>1 This bit in the DSPI_C serialized output frame will use the respective SIU_DSPICHL bit. |
| 5<br>DSPICHL5 | SIU_DSPICHL Select. Select one bit of the DSPI_C GPO Mask Output High Register to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective SIU_DSPICHL bit.<br>1 This bit in the DSPI_C serialized output frame will use the respective SIU_DSPICHL bit. |
| 6<br>DSPICHL6 | SIU_DSPICHL Select. Select one bit of the DSPI_C GPO Mask Output High Register to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective SIU_DSPICHL bit.<br>1 This bit in the DSPI_C serialized output frame will use the respective SIU_DSPICHL bit. |
| 7<br>DSPICHL7 | SIU_DSPICHL Select. Select one bit of the DSPI_C GPO Mask Output High Register to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective SIU_DSPICHL bit.<br>1 This bit in the DSPI_C serialized output frame will use the respective SIU_DSPICHL bit. |
| 8<br>DSPICHL8 | SIU_DSPICHL Select. Select one bit of the DSPI_C GPO Mask Output High Register to source the serialized DSPI_C output.                                                                                                                                                                                                   |

Table continues on the next page...

**SIU\_DSPICHL C field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                                                                           |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | 0 This bit in the DSPI_C serialized output frame will not use the respective SIU_DSPICH bit.<br>1 This bit in the DSPI_C serialized output frame will use the respective SIU_DSPICH bit.                                                                                                                              |
| 9<br>DSPICH9   | SIU_DSPICH Select. Select one bit of the DSPI_C GPO Mask Output High Register to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective SIU_DSPICH bit.<br>1 This bit in the DSPI_C serialized output frame will use the respective SIU_DSPICH bit. |
| 10<br>DSPICH10 | SIU_DSPICH Select. Select one bit of the DSPI_C GPO Mask Output High Register to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective SIU_DSPICH bit.<br>1 This bit in the DSPI_C serialized output frame will use the respective SIU_DSPICH bit. |
| 11<br>DSPICH11 | SIU_DSPICH Select. Select one bit of the DSPI_C GPO Mask Output High Register to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective SIU_DSPICH bit.<br>1 This bit in the DSPI_C serialized output frame will use the respective SIU_DSPICH bit. |
| 12<br>DSPICH12 | SIU_DSPICH Select. Select one bit of the DSPI_C GPO Mask Output High Register to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective SIU_DSPICH bit.<br>1 This bit in the DSPI_C serialized output frame will use the respective SIU_DSPICH bit. |
| 13<br>DSPICH13 | SIU_DSPICH Select. Select one bit of the DSPI_C GPO Mask Output High Register to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective SIU_DSPICH bit.<br>1 This bit in the DSPI_C serialized output frame will use the respective SIU_DSPICH bit. |
| 14<br>DSPICH14 | SIU_DSPICH Select. Select one bit of the DSPI_C GPO Mask Output High Register to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective SIU_DSPICH bit.<br>1 This bit in the DSPI_C serialized output frame will use the respective SIU_DSPICH bit. |
| 15<br>DSPICH15 | SIU_DSPICH Select. Select one bit of the DSPI_C GPO Mask Output High Register to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective SIU_DSPICH bit.<br>1 This bit in the DSPI_C serialized output frame will use the respective SIU_DSPICH bit. |
| 16<br>DSPICL16 | SIU_DSPICL Select. Select one bit of the DSPI_C GPO Mask Output Low Register to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective SIU_DSPICL bit.<br>1 This bit in the DSPI_C serialized output frame will use the respective SIU_DSPICL bit.  |
| 17<br>DSPICL17 | SIU_DSPICL Select. Select one bit of the DSPI_C GPO Mask Output Low Register to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective SIU_DSPICL bit.<br>1 This bit in the DSPI_C serialized output frame will use the respective SIU_DSPICL bit.  |
| 18<br>DSPICL18 | SIU_DSPICL Select. Select one bit of the DSPI_C GPO Mask Output Low Register to source the serialized DSPI_C output.                                                                                                                                                                                                  |

*Table continues on the next page...*

**SIU\_DSPICHL field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                                                                          |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | 0 This bit in the DSPI_C serialized output frame will not use the respective SIU_DSPICL bit.<br>1 This bit in the DSPI_C serialized output frame will use the respective SIU_DSPICL bit.                                                                                                                             |
| 19<br>DSPICL19 | SIU_DSPICL Select. Select one bit of the DSPI_C GPO Mask Output Low Register to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective SIU_DSPICL bit.<br>1 This bit in the DSPI_C serialized output frame will use the respective SIU_DSPICL bit. |
| 20<br>DSPICL20 | SIU_DSPICL Select. Select one bit of the DSPI_C GPO Mask Output Low Register to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective SIU_DSPICL bit.<br>1 This bit in the DSPI_C serialized output frame will use the respective SIU_DSPICL bit. |
| 21<br>DSPICL21 | SIU_DSPICL Select. Select one bit of the DSPI_C GPO Mask Output Low Register to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective SIU_DSPICL bit.<br>1 This bit in the DSPI_C serialized output frame will use the respective SIU_DSPICL bit. |
| 22<br>DSPICL22 | SIU_DSPICL Select. Select one bit of the DSPI_C GPO Mask Output Low Register to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective SIU_DSPICL bit.<br>1 This bit in the DSPI_C serialized output frame will use the respective SIU_DSPICL bit. |
| 23<br>DSPICL23 | SIU_DSPICL Select. Select one bit of the DSPI_C GPO Mask Output Low Register to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective SIU_DSPICL bit.<br>1 This bit in the DSPI_C serialized output frame will use the respective SIU_DSPICL bit. |
| 24<br>DSPICL24 | SIU_DSPICL Select. Select one bit of the DSPI_C GPO Mask Output Low Register to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective SIU_DSPICL bit.<br>1 This bit in the DSPI_C serialized output frame will use the respective SIU_DSPICL bit. |
| 25<br>DSPICL25 | SIU_DSPICL Select. Select one bit of the DSPI_C GPO Mask Output Low Register to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective SIU_DSPICL bit.<br>1 This bit in the DSPI_C serialized output frame will use the respective SIU_DSPICL bit. |
| 26<br>DSPICL26 | SIU_DSPICL Select. Select one bit of the DSPI_C GPO Mask Output Low Register to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective SIU_DSPICL bit.<br>1 This bit in the DSPI_C serialized output frame will use the respective SIU_DSPICL bit. |
| 27<br>DSPICL27 | SIU_DSPICL Select. Select one bit of the DSPI_C GPO Mask Output Low Register to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective SIU_DSPICL bit.<br>1 This bit in the DSPI_C serialized output frame will use the respective SIU_DSPICL bit. |
| 28<br>DSPICL28 | SIU_DSPICL Select. Select one bit of the DSPI_C GPO Mask Output Low Register to source the serialized DSPI_C output.                                                                                                                                                                                                 |

*Table continues on the next page...*

**SIU\_DSPICHL field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                                                                          |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | 0 This bit in the DSPI_C serialized output frame will not use the respective SIU_DSPICL bit.<br>1 This bit in the DSPI_C serialized output frame will use the respective SIU_DSPICL bit.                                                                                                                             |
| 29<br>DSPICL29 | SIU_DSPICL Select. Select one bit of the DSPI_C GPO Mask Output Low Register to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective SIU_DSPICL bit.<br>1 This bit in the DSPI_C serialized output frame will use the respective SIU_DSPICL bit. |
| 30<br>DSPICL30 | SIU_DSPICL Select. Select one bit of the DSPI_C GPO Mask Output Low Register to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective SIU_DSPICL bit.<br>1 This bit in the DSPI_C serialized output frame will use the respective SIU_DSPICL bit. |
| 31<br>DSPICL31 | SIU_DSPICL Select. Select one bit of the DSPI_C GPO Mask Output Low Register to source the serialized DSPI_C output.<br><br>0 This bit in the DSPI_C serialized output frame will not use the respective SIU_DSPICL bit.<br>1 This bit in the DSPI_C serialized output frame will use the respective SIU_DSPICL bit. |

**8.2.61 Serialized Output Signal Selection Register for DSPI\_C (SIU\_ETPUBC)**

Each bit in the DSPI serialized output frame can contain a signal routed from the output of one of three on-chip sources. The sources on this device are the eTPU module, eMIOS module, or a software-updated GPO data register. The mapping between module or data register and the output frame for a single DSPI channel is performed by independent 32-bit registers, one for each module plus one for the data register.

The following register is used by DSPI\_C to select the sources of the serialized output when running in DSI or CSI configuration.

Each register bit enables a path from the eTPU\_A/B channel, eMIOS channel and data register bit SIU\_DSPICH/SIU\_DSPICL to the equivalent bit position in the DSPI\_C serialized output frame. The user must ensure that bit selections from each of these registers do not overlap. Multiple sources are logically ORed, which provides the potential for combining outputs from multiple timer channels and data registers to produce more complex bit behavior.

Address: 0h base + D6Ch offset = D6Ch

| Bit   | 0       | 1       | 2       | 3       | 4       | 5       | 6      | 7      | 8      | 9      | 10     | 11     | 12     | 13     | 14     | 15     |
|-------|---------|---------|---------|---------|---------|---------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| R     | ETPUB15 | ETPUB14 | ETPUB13 | ETPUB12 | ETPUB11 | ETPUB10 | ETPUB9 | ETPUB8 | ETPUB7 | ETPUB6 | ETPUB5 | ETPUB4 | ETPUB3 | ETPUB2 | ETPUB1 | ETPUB0 |
| W     | ETPUB15 | ETPUB14 | ETPUB13 | ETPUB12 | ETPUB11 | ETPUB10 | ETPUB9 | ETPUB8 | ETPUB7 | ETPUB6 | ETPUB5 | ETPUB4 | ETPUB3 | ETPUB2 | ETPUB1 | ETPUB0 |
| Reset | 0       | 0       | 0       | 0       | 0       | 0       | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      |

## Memory Map and Register Definition

|       |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
|-------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| Bit   | 16      | 17      | 18      | 19      | 20      | 21      | 22      | 23      | 24      | 25      | 26      | 27      | 28      | 29      | 30      | 31      |
| R     | ETPUB31 | ETPUB30 | ETPUB29 | ETPUB28 | ETPUB27 | ETPUB26 | ETPUB25 | ETPUB24 | ETPUB23 | ETPUB22 | ETPUB21 | ETPUB20 | ETPUB19 | ETPUB18 | ETPUB17 | ETPUB16 |
| W     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| Reset | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       |

### SIU\_ETPUBC field descriptions

| Field        | Description                                                                                                                                                                                                                                                                         |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>ETPUB15 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_B channel. |
| 1<br>ETPUB14 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_B channel. |
| 2<br>ETPUB13 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_B channel. |
| 3<br>ETPUB12 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_B channel. |
| 4<br>ETPUB11 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_B channel. |
| 5<br>ETPUB10 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_B channel. |
| 6<br>ETPUB9  | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_B channel. |
| 7<br>ETPUB8  | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_B channel. |
| 8<br>ETPUB7  | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_B channel. |
| 9<br>ETPUB6  | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_B channel. |

Table continues on the next page...

**SIU\_ETPUBC field descriptions (continued)**

| Field         | Description                                                                                                                                                                                                                                                                         |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 10<br>ETPUB5  | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_B channel. |
| 11<br>ETPUB4  | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_B channel. |
| 12<br>ETPUB3  | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_B channel. |
| 13<br>ETPUB2  | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_B channel. |
| 14<br>ETPUB1  | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_B channel. |
| 15<br>ETPUB0  | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_B channel. |
| 16<br>ETPUB31 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_B channel. |
| 17<br>ETPUB30 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_B channel. |
| 18<br>ETPUB29 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_B channel. |
| 19<br>ETPUB28 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_B channel. |
| 20<br>ETPUB27 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_B channel. |
| 21<br>ETPUB26 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_B channel. |

*Table continues on the next page...*

**SIU\_ETPUBC field descriptions (continued)**

| Field         | Description                                                                                                                                                                                                                                                                         |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 22<br>ETPUB25 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_B channel. |
| 23<br>ETPUB24 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_B channel. |
| 24<br>ETPUB23 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_B channel. |
| 25<br>ETPUB22 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_B channel. |
| 26<br>ETPUB21 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_B channel. |
| 27<br>ETPUB20 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_B channel. |
| 28<br>ETPUB19 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_B channel. |
| 29<br>ETPUB18 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_B channel. |
| 30<br>ETPUB17 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_B channel. |
| 31<br>ETPUB16 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_C output.<br>0 This bit in the DSPI_C serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_C serialized output frame will use the respective eTPU_B channel. |



## 8.2.62 Serialized Output Signal Selection Register for DSPI\_D (SIU\_ETPUBD)

Each bit in the DSPI serialized output frame can contain a signal routed from the output of one of three on-chip sources. The sources on this device are the eTPU module, eMIOS module, or a software-updated GPO data register. The mapping between module or data register and the output frame for a single DSPI channel is performed by independent 32-bit registers, one for each module plus one for the data register.

The following register is used by DSPI\_D to select the sources of the serialized output when running in DSI or CSI configuration.

The register bit enables a path from the eTPU\_B channel or eMIOS channel to the equivalent bit position in the DSPI\_D serialized output frame. The user must ensure that bit selections from each of these registers do not overlap. Multiple sources are logically ORed, which provides the potential for combining outputs from multiple timer channels and data registers to produce more complex bit behavior.

Address: 0h base + D70h offset = D70h

|       |         |         |         |         |         |         |    |    |    |    |         |         |         |         |         |         |
|-------|---------|---------|---------|---------|---------|---------|----|----|----|----|---------|---------|---------|---------|---------|---------|
| Bit   | 0       | 1       | 2       | 3       | 4       | 5       | 6  | 7  | 8  | 9  | 10      | 11      | 12      | 13      | 14      | 15      |
| R     | ETPUB21 | ETPUB20 | ETPUB19 | ETPUB18 | ETPUB17 | ETPUB16 | 0  |    |    |    | ETPUB29 | ETPUB28 | ETPUB27 | ETPUB26 | ETPUB25 | ETPUB24 |
| W     |         |         |         |         |         |         |    |    |    |    |         |         |         |         |         |         |
| Reset | 0       | 0       | 0       | 0       | 0       | 0       | 0  | 0  | 0  | 0  | 0       | 0       | 0       | 0       | 0       | 0       |
| Bit   | 16      | 17      | 18      | 19      | 20      | 21      | 22 | 23 | 24 | 25 | 26      | 27      | 28      | 29      | 30      | 31      |
| R     | 0       |         |         |         |         |         |    |    |    |    |         |         |         |         |         |         |
| W     |         |         |         |         |         |         |    |    |    |    |         |         |         |         |         |         |
| Reset | 0       | 0       | 0       | 0       | 0       | 0       | 0  | 0  | 0  | 0  | 0       | 0       | 0       | 0       | 0       | 0       |

**SIU\_ETPUBD field descriptions**

| Field        | Description                                                                                                                                                                                                                                                                         |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>ETPUB21 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_D output.<br>0 This bit in the DSPI_D serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_D serialized output frame will use the respective eTPU_B channel. |
| 1<br>ETPUB20 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_D output.<br>0 This bit in the DSPI_D serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_D serialized output frame will use the respective eTPU_B channel. |
| 2<br>ETPUB19 | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_D output.                                                                                                                                                                                             |

*Table continues on the next page...*

## SIU\_ETPUBD field descriptions (continued)

| Field             | Description                                                                                                                                                                                                                                                                             |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | 0 This bit in the DSPI_D serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_D serialized output frame will use the respective eTPU_B channel.                                                                                                |
| 3<br>ETPUB18      | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_D output.<br><br>0 This bit in the DSPI_D serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_D serialized output frame will use the respective eTPU_B channel. |
| 4<br>ETPUB17      | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_D output.<br><br>0 This bit in the DSPI_D serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_D serialized output frame will use the respective eTPU_B channel. |
| 5<br>ETPUB16      | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_D output.<br><br>0 This bit in the DSPI_D serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_D serialized output frame will use the respective eTPU_B channel. |
| 6–9<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                 |
| 10<br>ETPUB29     | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_D output.<br><br>0 This bit in the DSPI_D serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_D serialized output frame will use the respective eTPU_B channel. |
| 11<br>ETPUB28     | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_D output.<br><br>0 This bit in the DSPI_D serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_D serialized output frame will use the respective eTPU_B channel. |
| 12<br>ETPUB27     | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_D output.<br><br>0 This bit in the DSPI_D serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_D serialized output frame will use the respective eTPU_B channel. |
| 13<br>ETPUB26     | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_D output.<br><br>0 This bit in the DSPI_D serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_D serialized output frame will use the respective eTPU_B channel. |
| 14<br>ETPUB25     | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_D output.<br><br>0 This bit in the DSPI_D serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_D serialized output frame will use the respective eTPU_B channel. |
| 15<br>ETPUB24     | eTPU_B Channel Select. Select an eTPU_B channel to source the serialized DSPI_D output.<br><br>0 This bit in the DSPI_D serialized output frame will not use the respective eTPU_B channel.<br>1 This bit in the DSPI_D serialized output frame will use the respective eTPU_B channel. |
| 16–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                 |

### 8.2.63 Serialized Output Signal Selection Register for DSPI\_D (SIU\_EMIOSD)

Each bit in the DSPI serialized output frame can contain a signal routed from the output of one of three on-chip sources. The sources on this device are the eTPU module, eMIOS module, or a software-updated GPO data register. The mapping between module or data register and the output frame for a single DSPI channel is performed by independent 32-bit registers, one for each module plus one for the data register.

The following register is used by DSPI\_D to select the sources of the serialized output when running in DSI or CSI configuration.

The register bit enables a path from the eTPU\_B channel or eMIOS channel to the equivalent bit position in the DSPI\_D serialized output frame. The user must ensure that bit selections from each of these registers do not overlap. Multiple sources are logically ORed, which provides the potential for combining outputs from multiple timer channels and data registers to produce more complex bit behavior.

Address: 0h base + D74h offset = D74h

|       |    |    |    |    |    |    |           |           |           |           |    |    |    |    |    |    |
|-------|----|----|----|----|----|----|-----------|-----------|-----------|-----------|----|----|----|----|----|----|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6         | 7         | 8         | 9         | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0  |    |    |    |    |    | EMIOS1_3_ | EMIOS1_2_ | EMIOS1_5_ | EMIOS1_4_ |    | 0  |    |    |    |    |
| W     |    |    |    |    |    |    | OUT6      | OUT7      | OUT8      | OUT9      |    |    |    |    |    |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0         | 0         | 0         | 0         | 0  | 0  | 0  | 0  | 0  | 0  |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22        | 23        | 24        | 25        | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0  |    |    |    |    |    |           |           |           |           |    |    |    |    |    |    |
| W     |    |    |    |    |    |    |           |           |           |           |    |    |    |    |    |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0         | 0         | 0         | 0         | 0  | 0  | 0  | 0  | 0  | 0  |

**SIU\_EMIOSD field descriptions**

| Field           | Description                                                                                                                                                                                    |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–5<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                        |
| 6<br>EMIOS1_3_  | eMIOS_1 Channel Select. Select an eMIOS_1 channel to source the serialized DSPI_D output.                                                                                                      |
| OUT6            | 0 This bit in the DSPI_D serialized output frame will not use the respective eMIOS_1 channel 3.<br>1 This bit in the DSPI_D serialized output frame will use the respective eMIOS_1 channel 3. |

*Table continues on the next page...*

## SIU\_EMIOSD field descriptions (continued)

| Field                  | Description                                                                                                                                                                                                                                                                                 |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7<br>EMIOS1_2_<br>OUT7 | eMIOS_1 Channel Select. Select an eMIOS_1 channel to source the serialized DSPI_D output.<br>0 This bit in the DSPI_D serialized output frame will not use the respective eMIOS_1 channel 2.<br>1 This bit in the DSPI_D serialized output frame will use the respective eMIOS_1 channel 2. |
| 8<br>EMIOS1_5_<br>OUT8 | eMIOS_1 Channel Select. Select an eMIOS_1 channel to source the serialized DSPI_D output.<br>0 This bit in the DSPI_D serialized output frame will not use the respective eMIOS_1 channel 5.<br>1 This bit in the DSPI_D serialized output frame will use the respective eMIOS_1 channel 5. |
| 9<br>EMIOS1_4_<br>OUT9 | eMIOS_1 Channel Select. Select an eMIOS_1 channel to source the serialized DSPI_D output.<br>0 This bit in the DSPI_D serialized output frame will not use the respective eMIOS_1 channel 4.<br>1 This bit in the DSPI_D serialized output frame will use the respective eMIOS_1 channel 4. |
| 10–31<br>Reserved      | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                     |

## 8.2.64 Serialized Output Signal Selection Register for DSPI\_D (SIU\_DSPIDHLD)

Address: 0h base + D78h offset = D78h

|       |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |
|-------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| Bit   | 0        | 1        | 2        | 3        | 4        | 5        | 6        | 7        | 8        | 9        | 10       | 11       | 12       | 13       | 14       | 15       |
| R     | DSPIDH0  | DSPIDH1  | DSPIDH2  | DSPIDH3  | DSPIDH4  | DSPIDH5  | DSPIDH6  | DSPIDH7  | DSPIDH8  | DSPIDH9  | DSPIDH10 | DSPIDH11 | DSPIDH12 | DSPIDH13 | DSPIDH14 | DSPIDH15 |
| W     |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |
| Reset | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        |
| Bit   | 16       | 17       | 18       | 19       | 20       | 21       | 22       | 23       | 24       | 25       | 26       | 27       | 28       | 29       | 30       | 31       |
| R     | DSPIDL16 | DSPIDL17 | DSPIDL18 | DSPIDL19 | DSPIDL20 | DSPIDL21 | DSPIDL22 | DSPIDL23 | DSPIDL24 | DSPIDL25 | DSPIDL26 | DSPIDL27 | DSPIDL28 | DSPIDL29 | DSPIDL30 | DSPIDL31 |
| W     |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |          |
| Reset | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        |

## SIU\_DSPIDHLD field descriptions

| Field        | Description                                                                                                                                                                                                                                                                                                       |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>DSPIDH0 | SIU_DSPIDH Select. Select one bit of the DSPI_D GPO Mask Output High Register to source the serialized DSPI_D output.<br>0 This bit in the DSPI_D serialized output frame will not use the respective SIU_DSPIDH bit.<br>1 This bit in the DSPI_D serialized output frame will use the respective SIU_DSPIDH bit. |
| 1<br>DSPIDH1 | SIU_DSPIDH Select. Select one bit of the DSPI_D GPO Mask Output High Register to source the serialized DSPI_D output.                                                                                                                                                                                             |

Table continues on the next page...

**SIU\_DSPIDHLD field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                                                                           |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | 0 This bit in the DSPI_D serialized output frame will not use the respective SIU_DSPIDH bit.<br>1 This bit in the DSPI_D serialized output frame will use the respective SIU_DSPIDH bit.                                                                                                                              |
| 2<br>DSPIDH2   | SIU_DSPIDH Select. Select one bit of the DSPI_D GPO Mask Output High Register to source the serialized DSPI_D output.<br><br>0 This bit in the DSPI_D serialized output frame will not use the respective SIU_DSPIDH bit.<br>1 This bit in the DSPI_D serialized output frame will use the respective SIU_DSPIDH bit. |
| 3<br>DSPIDH3   | SIU_DSPIDH Select. Select one bit of the DSPI_D GPO Mask Output High Register to source the serialized DSPI_D output.<br><br>0 This bit in the DSPI_D serialized output frame will not use the respective SIU_DSPIDH bit.<br>1 This bit in the DSPI_D serialized output frame will use the respective SIU_DSPIDH bit. |
| 4<br>DSPIDH4   | SIU_DSPIDH Select. Select one bit of the DSPI_D GPO Mask Output High Register to source the serialized DSPI_D output.<br><br>0 This bit in the DSPI_D serialized output frame will not use the respective SIU_DSPIDH bit.<br>1 This bit in the DSPI_D serialized output frame will use the respective SIU_DSPIDH bit. |
| 5<br>DSPIDH5   | SIU_DSPIDH Select. Select one bit of the DSPI_D GPO Mask Output High Register to source the serialized DSPI_D output.<br><br>0 This bit in the DSPI_D serialized output frame will not use the respective SIU_DSPIDH bit.<br>1 This bit in the DSPI_D serialized output frame will use the respective SIU_DSPIDH bit. |
| 6<br>DSPIDH6   | SIU_DSPIDH Select. Select one bit of the DSPI_D GPO Mask Output High Register to source the serialized DSPI_D output.<br><br>0 This bit in the DSPI_D serialized output frame will not use the respective SIU_DSPIDH bit.<br>1 This bit in the DSPI_D serialized output frame will use the respective SIU_DSPIDH bit. |
| 7<br>DSPIDH7   | SIU_DSPIDH Select. Select one bit of the DSPI_D GPO Mask Output High Register to source the serialized DSPI_D output.<br><br>0 This bit in the DSPI_D serialized output frame will not use the respective SIU_DSPIDH bit.<br>1 This bit in the DSPI_D serialized output frame will use the respective SIU_DSPIDH bit. |
| 8<br>DSPIDH8   | SIU_DSPIDH Select. Select one bit of the DSPI_D GPO Mask Output High Register to source the serialized DSPI_D output.<br><br>0 This bit in the DSPI_D serialized output frame will not use the respective SIU_DSPIDH bit.<br>1 This bit in the DSPI_D serialized output frame will use the respective SIU_DSPIDH bit. |
| 9<br>DSPIDH9   | SIU_DSPIDH Select. Select one bit of the DSPI_D GPO Mask Output High Register to source the serialized DSPI_D output.<br><br>0 This bit in the DSPI_D serialized output frame will not use the respective SIU_DSPIDH bit.<br>1 This bit in the DSPI_D serialized output frame will use the respective SIU_DSPIDH bit. |
| 10<br>DSPIDH10 | SIU_DSPIDH Select. Select one bit of the DSPI_D GPO Mask Output High Register to source the serialized DSPI_D output.<br><br>0 This bit in the DSPI_D serialized output frame will not use the respective SIU_DSPIDH bit.<br>1 This bit in the DSPI_D serialized output frame will use the respective SIU_DSPIDH bit. |
| 11<br>DSPIDH11 | SIU_DSPIDH Select. Select one bit of the DSPI_D GPO Mask Output High Register to source the serialized DSPI_D output.                                                                                                                                                                                                 |

*Table continues on the next page...*

**SIU\_DSPIDHLD field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                                                                           |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | 0 This bit in the DSPI_D serialized output frame will not use the respective SIU_DSPIDH bit.<br>1 This bit in the DSPI_D serialized output frame will use the respective SIU_DSPIDH bit.                                                                                                                              |
| 12<br>DSPIDH12 | SIU_DSPIDH Select. Select one bit of the DSPI_D GPO Mask Output High Register to source the serialized DSPI_D output.<br><br>0 This bit in the DSPI_D serialized output frame will not use the respective SIU_DSPIDH bit.<br>1 This bit in the DSPI_D serialized output frame will use the respective SIU_DSPIDH bit. |
| 13<br>DSPIDH13 | SIU_DSPIDH Select. Select one bit of the DSPI_D GPO Mask Output High Register to source the serialized DSPI_D output.<br><br>0 This bit in the DSPI_D serialized output frame will not use the respective SIU_DSPIDH bit.<br>1 This bit in the DSPI_D serialized output frame will use the respective SIU_DSPIDH bit. |
| 14<br>DSPIDH14 | SIU_DSPIDH Select. Select one bit of the DSPI_D GPO Mask Output High Register to source the serialized DSPI_D output.<br><br>0 This bit in the DSPI_D serialized output frame will not use the respective SIU_DSPIDH bit.<br>1 This bit in the DSPI_D serialized output frame will use the respective SIU_DSPIDH bit. |
| 15<br>DSPIDH15 | SIU_DSPIDH Select. Select one bit of the DSPI_D GPO Mask Output High Register to source the serialized DSPI_D output.<br><br>0 This bit in the DSPI_D serialized output frame will not use the respective SIU_DSPIDH bit.<br>1 This bit in the DSPI_D serialized output frame will use the respective SIU_DSPIDH bit. |
| 16<br>DSPIDL16 | SIU_DSPIDL Select. Select one bit of the DSPI_D GPO Mask Output Low Register to source the serialized DSPI_D output.<br><br>0 This bit in the DSPI_D serialized output frame will not use the respective SIU_DSPIDL bit.<br>1 This bit in the DSPI_D serialized output frame will use the respective SIU_DSPIDL bit.  |
| 17<br>DSPIDL17 | SIU_DSPIDL Select. Select one bit of the DSPI_D GPO Mask Output Low Register to source the serialized DSPI_D output.<br><br>0 This bit in the DSPI_D serialized output frame will not use the respective SIU_DSPIDL bit.<br>1 This bit in the DSPI_D serialized output frame will use the respective SIU_DSPIDL bit.  |
| 18<br>DSPIDL18 | SIU_DSPIDL Select. Select one bit of the DSPI_D GPO Mask Output Low Register to source the serialized DSPI_D output.<br><br>0 This bit in the DSPI_D serialized output frame will not use the respective SIU_DSPIDL bit.<br>1 This bit in the DSPI_D serialized output frame will use the respective SIU_DSPIDL bit.  |
| 19<br>DSPIDL19 | SIU_DSPIDL Select. Select one bit of the DSPI_D GPO Mask Output Low Register to source the serialized DSPI_D output.<br><br>0 This bit in the DSPI_D serialized output frame will not use the respective SIU_DSPIDL bit.<br>1 This bit in the DSPI_D serialized output frame will use the respective SIU_DSPIDL bit.  |
| 20<br>DSPIDL20 | SIU_DSPIDL Select. Select one bit of the DSPI_D GPO Mask Output Low Register to source the serialized DSPI_D output.<br><br>0 This bit in the DSPI_D serialized output frame will not use the respective SIU_DSPIDL bit.<br>1 This bit in the DSPI_D serialized output frame will use the respective SIU_DSPIDL bit.  |
| 21<br>DSPIDL21 | SIU_DSPIDL Select. Select one bit of the DSPI_D GPO Mask Output Low Register to source the serialized DSPI_D output.                                                                                                                                                                                                  |

*Table continues on the next page...*

**SIU\_DSPIDHLD field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                                                                          |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | 0 This bit in the DSPI_D serialized output frame will not use the respective SIU_DSPIDL bit.<br>1 This bit in the DSPI_D serialized output frame will use the respective SIU_DSPIDL bit.                                                                                                                             |
| 22<br>DSPIDL22 | SIU_DSPIDL Select. Select one bit of the DSPI_D GPO Mask Output Low Register to source the serialized DSPI_D output.<br><br>0 This bit in the DSPI_D serialized output frame will not use the respective SIU_DSPIDL bit.<br>1 This bit in the DSPI_D serialized output frame will use the respective SIU_DSPIDL bit. |
| 23<br>DSPIDL23 | SIU_DSPIDL Select. Select one bit of the DSPI_D GPO Mask Output Low Register to source the serialized DSPI_D output.<br><br>0 This bit in the DSPI_D serialized output frame will not use the respective SIU_DSPIDL bit.<br>1 This bit in the DSPI_D serialized output frame will use the respective SIU_DSPIDL bit. |
| 24<br>DSPIDL24 | SIU_DSPIDL Select. Select one bit of the DSPI_D GPO Mask Output Low Register to source the serialized DSPI_D output.<br><br>0 This bit in the DSPI_D serialized output frame will not use the respective SIU_DSPIDL bit.<br>1 This bit in the DSPI_D serialized output frame will use the respective SIU_DSPIDL bit. |
| 25<br>DSPIDL25 | SIU_DSPIDL Select. Select one bit of the DSPI_D GPO Mask Output Low Register to source the serialized DSPI_D output.<br><br>0 This bit in the DSPI_D serialized output frame will not use the respective SIU_DSPIDL bit.<br>1 This bit in the DSPI_D serialized output frame will use the respective SIU_DSPIDL bit. |
| 26<br>DSPIDL26 | SIU_DSPIDL Select. Select one bit of the DSPI_D GPO Mask Output Low Register to source the serialized DSPI_D output.<br><br>0 This bit in the DSPI_D serialized output frame will not use the respective SIU_DSPIDL bit.<br>1 This bit in the DSPI_D serialized output frame will use the respective SIU_DSPIDL bit. |
| 27<br>DSPIDL27 | SIU_DSPIDL Select. Select one bit of the DSPI_D GPO Mask Output Low Register to source the serialized DSPI_D output.<br><br>0 This bit in the DSPI_D serialized output frame will not use the respective SIU_DSPIDL bit.<br>1 This bit in the DSPI_D serialized output frame will use the respective SIU_DSPIDL bit. |
| 28<br>DSPIDL28 | SIU_DSPIDL Select. Select one bit of the DSPI_D GPO Mask Output Low Register to source the serialized DSPI_D output.<br><br>0 This bit in the DSPI_D serialized output frame will not use the respective SIU_DSPIDL bit.<br>1 This bit in the DSPI_D serialized output frame will use the respective SIU_DSPIDL bit. |
| 29<br>DSPIDL29 | SIU_DSPIDL Select. Select one bit of the DSPI_D GPO Mask Output Low Register to source the serialized DSPI_D output.<br><br>0 This bit in the DSPI_D serialized output frame will not use the respective SIU_DSPIDL bit.<br>1 This bit in the DSPI_D serialized output frame will use the respective SIU_DSPIDL bit. |
| 30<br>DSPIDL30 | SIU_DSPIDL Select. Select one bit of the DSPI_D GPO Mask Output Low Register to source the serialized DSPI_D output.<br><br>0 This bit in the DSPI_D serialized output frame will not use the respective SIU_DSPIDL bit.<br>1 This bit in the DSPI_D serialized output frame will use the respective SIU_DSPIDL bit. |
| 31<br>DSPIDL31 | SIU_DSPIDL Select. Select one bit of the DSPI_D GPO Mask Output Low Register to source the serialized DSPI_D output.                                                                                                                                                                                                 |

*Table continues on the next page...*

**SIU\_DSPIDHLD field descriptions (continued)**

| Field | Description                                                                                |
|-------|--------------------------------------------------------------------------------------------|
| 0     | This bit in the DSPI_D serialized output frame will not use the respective SIU_DSPIDL bit. |
| 1     | This bit in the DSPI_D serialized output frame will use the respective SIU_DSPIDL bit.     |

**8.2.65 GPIO Pin Data Input Register (SIU\_GPDIn)**

Previous devices are limited to supporting a maximum of 256 general purpose input registers, and thus a maximum GPIO input assignment of GPIO[0:255]. The reason was that only 256 bytes of SIU address space were allocated for the GPIO Pin Data Input registers, and adjacent memory locations are unavailable. To accommodate the extended number of the GPIO pads, this device has a new memory map definition of 512 bytes for allocating up to 512 GPIO Pin Data Input registers. For compatibility, the first used 256 GPIO Pin Data Input registers are mapped to both the old and new locations.

Each SIU\_GPDIn register specifies the input state for the GPDIn[n] pin. The *n* notation in the SIU\_GPDIn register name relates to the *[n]* in GPIO[n] signal name. For example, SIU\_GPDIn246 contains the PDI246 bit for CNTXD\_GPIO246. Gaps exist in the memory map where registers correspond to GPIO pins that are not implemented in the package.

Software reads the SIU\_GPDIn registers to get the input state of the external GPIO pin. Each GPDIn register contains the input state of one external GPIO pin. If a GPDIn register is configured as output, and the input buffer enable bit is set to one in the PCR register, the SIU\_GPDIn register reflects the state of the output pin.

**Table 8-1183. Implemented GPIO Pin Data Input Registers**

| Offset          | GPIO register                         |
|-----------------|---------------------------------------|
| 0x0E00 - 0x0E4A | Reserved                              |
| 0x0E4B - 0x0E6E | GPIO Pin Data Input Registers 75-110  |
| 0x0E6F - 0x0E70 | Reserved                              |
| 0x0E71 - 0x0ECC | GPIO Pin Data Input Registers 113-204 |
| 0x0ECD - 0x0ECF | Reserved                              |
| 0x0ED0 - 0x0ED5 | GPIO Pin Data Input Registers 208-213 |
| 0x0ED6 - 0x0EDB | Reserved                              |
| 0x0EDC - 0x0EDF | GPIO Pin Data Input Registers 220-223 |
| 0x0EE0 - 0x0EE6 | Reserved                              |
| 0x0EE7 - 0x0EFD | GPIO Pin Data Input Registers 231-253 |
| 0x0EFE - 0x0EFF | Reserved                              |
| 0x0F00 - 0x0F33 | GPIO Pin Data Input Registers 256-307 |
| 0x0F34 - 0x0FAF | Reserved                              |
| 0x0FB0 - 0x0FB5 | GPIO Pin Data Input Registers 432-437 |

*Table continues on the next page...*



**Table 8-1183. Implemented GPIO Pin Data Input Registers (continued)**

| Offset          | GPIO register                         |
|-----------------|---------------------------------------|
| 0x0FB6 - 0x0FB7 | Reserved                              |
| 0x0FB8 - 0x0FBA | GPIO Pin Data Input Registers 440-442 |
| 0x0FBB          | GPIO Pin Data Register 443            |
| 0x0FBC          | GPIO Pin Data Register 444            |
| 0x0FBD - 0x0FC1 | GPIO Pin Data Registers 445-449       |
| 0x0FC2          | GPIO Pin Data Register 450            |
| 0x0FC3 - 0x0FD8 | GPIO Pin Data Registers 451-472       |
| 0x0FD9          | Reserved                              |
| 0x0FDA - 0x0FEB | GPIO Pin Data Registers 474-491       |
| 0x0FEC - 0x0FFF | Reserved                              |

Address: 0h base + E4Bh offset + (1d × i), where i=0d to 416d

| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6    | 7 |
|-------|---|---|---|---|---|---|------|---|
| Read  | 0 |   |   |   |   |   | PDIn |   |
| Write |   |   |   |   |   |   |      |   |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0    | * |

\* Notes:

- PDIn field: The value depends on what is being driven on the corresponding pin.

### SIU\_GPDIn field descriptions

| Field           | Description                                                                                                                                                                     |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–6<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                         |
| 7<br>PDIn       | Pin Data In. This bit reflects the state on the external GPIO/GPI pin. If PCR <sub>n</sub> [IBE] is 1, then:<br>0 Signal on pin is logic zero.<br>1 Signal on pin is logic one. |

## 8.2.66 Input Multiplexing Register0 (SIU\_IMUX0)

The SIU\_IMUX registers specify the source of peripheral inputs that may come from multiple package pins. The registers allow selection from 4 sources for each input. Each SIU\_IMUX register contains 16 inputs, each with 2 bits (4 sources).

Some registers are reserved for future use, and some fields within some of the registers are also reserved.

## Memory Map and Register Definition

Address: 0h base + 1200h offset = 1200h

|       |   |   |   |   |   |   |   |   |          |   |          |    |         |    |         |    |
|-------|---|---|---|---|---|---|---|---|----------|---|----------|----|---------|----|---------|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8        | 9 | 10       | 11 | 12      | 13 | 14      | 15 |
| R     | 0 |   |   |   |   |   |   |   | MUXSEL11 |   | MUXSEL10 |    | MUXSEL9 |    | MUXSEL8 |    |
| W     |   |   |   |   |   |   |   |   |          |   |          |    |         |    |         |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0        | 0 | 0        | 0  | 0       | 0  | 0       | 0  |

|       |         |    |         |    |         |    |         |    |         |    |         |    |         |    |         |    |
|-------|---------|----|---------|----|---------|----|---------|----|---------|----|---------|----|---------|----|---------|----|
| Bit   | 16      | 17 | 18      | 19 | 20      | 21 | 22      | 23 | 24      | 25 | 26      | 27 | 28      | 29 | 30      | 31 |
| R     | MUXSEL7 |    | MUXSEL6 |    | MUXSEL5 |    | MUXSEL4 |    | MUXSEL3 |    | MUXSEL2 |    | MUXSEL1 |    | MUXSEL0 |    |
| W     |         |    |         |    |         |    |         |    |         |    |         |    |         |    |         |    |
| Reset | 0       | 0  | 0       | 0  | 0       | 0  | 0       | 0  | 0       | 0  | 0       | 0  | 0       | 0  | 0       | 0  |

### SIU\_IMUX0 field descriptions

| Field             | Description                                                                                                                                                                                                                                                       |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–7<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                           |
| 8–9<br>MUXSEL11   | Selects the source pad for SENT Receiver B Channel 5 Input : SENT5_B.<br><br>00 Selects input from "ETPUC10_IRQ1_SENT5_B_GPIO451" pad (PCR: 451).<br>01 Selects input from "PCSB5_PCSC0_SENT5_B_FEC_RXD2_GPIO110" pad (PCR: 110).<br>10 Reserved<br>11 Reserved   |
| 10–11<br>MUXSEL10 | Selects the source pad for SENT Receiver B Channel 4 Input : SENT4_B.<br><br>00 Selects input from "ETPUC9_IRQ0_SENT4_B_ANB44_GPI450" pad (PCR: 450).<br>01 Selects input from "PCSB4_SCKC_SENT4_B_FEC_MDC_GPIO109" pad (PCR: 109).<br>10 Reserved<br>11 Reserved |
| 12–13<br>MUXSEL9  | Selects the source pad for SENT Receiver B Channel 3 Input : SENT3_B.<br><br>00 Selects input from "ETPUC8_PCSE5_SENT3_B_GPIO449" pad (PCR: 449).<br>01 Selects input from "PCSB3_SINC_SENT3_B_GPIO108" pad (PCR: 108).<br>10 Reserved<br>11 Reserved             |
| 14–15<br>MUXSEL8  | Selects the source pad for SENT Receiver B Channel 2 Input : SENT2_B.<br><br>00 Selects input from "ETPUC7_PCSE4_SENT2_B_GPIO448" pad (PCR: 448).<br>01 Selects input from "PCSB2_SOUTC_SENT2_B_FEC_RXD3_GPIO107" pad (PCR: 107).<br>10 Reserved<br>11 Reserved   |
| 16–17<br>MUXSEL7  | Selects the source pad for SENT Receiver B Channel 1 Input : SENT1_B.<br><br>00 Selects input from "ETPUC6_PCSE3_SENT1_B_GPIO447" pad (PCR: 447).<br>01 Selects input from "SCKA_PCSC1_MCANTX0_SENT1_B_GPIO93" pad (PCR: 93).<br>10 Reserved<br>11 Reserved       |
| 18–19<br>MUXSEL6  | Selects the source pad for SENT Receiver B Channel 0 Input : SENT0_B.<br><br>00 Selects input from "ETPUC5_PCSE2_SENT0_B_GPIO446" pad (PCR: 446).<br>01 Selects input from "FEC_TXCLK_REFCLK_SENT0_B_GPI474" pad (PCR: 474).<br>10 Reserved<br>11 Reserved        |

Table continues on the next page...

**SIU\_IMUX0 field descriptions (continued)**

| Field            | Description                                                                                                                                                                                                                                                        |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 20–21<br>MUXSEL5 | <p>Selects the source pad for SENT Receiver A Channel 5 Input : SENT5_A.</p> <p>00 Selects input from "ETPUC4_SENT5_A_TCRCLKC_GPIO445" pad (PCR: 445).<br/> 01 Selects input from "FEC_RXD1_SENT5_A_GPIO253" pad (PCR: 253).<br/> 10 Reserved<br/> 11 Reserved</p> |
| 22–23<br>MUXSEL4 | <p>Selects the source pad for SENT Receiver A Channel 4 Input : SENT4_A.</p> <p>00 Selects input from "ETPUC3_SENT4_A_ANB45_GPI444" pad (PCR: 444).<br/> 01 Selects input from "FEC_TX_EN_SENT4_A_GPIO252" pad (PCR: 252).<br/> 10 Reserved<br/> 11 Reserved</p>   |
| 24–25<br>MUXSEL3 | <p>Selects the source pad for SENT Receiver A Channel 3 Input : SENT3_A.</p> <p>00 Selects input from "ETPUC2_SENT3_A_GPIO443" pad (PCR: 443).<br/> 01 Selects input from "FEC_TXD1_SENT3_A_GPIO251" pad (PCR: 251).<br/> 10 Reserved<br/> 11 Reserved</p>         |
| 26–27<br>MUXSEL2 | <p>Selects the source pad for SENT Receiver A Channel 2 Input : SENT2_A.</p> <p>00 Selects input from "SENT2_A_ANB43_GPI475" pad (PCR: 475).<br/> 01 Selects input from "FEC_RXD0_SENT2_A_GPIO250" pad (PCR: 250).<br/> 10 Reserved<br/> 11 Reserved</p>           |
| 28–29<br>MUXSEL1 | <p>Selects the source pad for SENT Receiver A Channel 1 Input : SENT1_A.</p> <p>00 Selects input from "ETPUC1_SENT1_A_ANB42_GPI442" pad (PCR: 442).<br/> 01 Selects input from "FEC_RX_DV_SENT1_A_GPIO249" pad (PCR: 249).<br/> 10 Reserved<br/> 11 Reserved</p>   |
| 30–31<br>MUXSEL0 | <p>Selects the source pad for SENT Receiver A Channel 0 Input : SENT0_A.</p> <p>00 Selects input from "ETPUC0_SENT0_A_ANB41_GPI441" pad (PCR: 441).<br/> 01 Selects input from "FEC_TXD0_SENT0_A_GPIO248" pad (PCR: 248).<br/> 10 Reserved<br/> 11 Reserved</p>    |

**8.2.67 Input Multiplexing Register1 (SIU\_IMUX1)**

The SIU\_IMUX registers specify the source of peripheral inputs that may come from multiple package pins. The registers allow selection from 4 sources for each input. Each SIU\_IMUX register contains 16 inputs, each with 2 bits (4 sources).

Some registers are reserved for future use, and some fields within some of the registers are also reserved.

## Memory Map and Register Definition

Address: 0h base + 1204h offset = 1204h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |    |    |    |         |    |    |    |    |    |    |    |         |    |    |         |  |
|-------|----|----|----|----|---------|----|----|----|----|----|----|----|---------|----|----|---------|--|
| Bit   | 16 | 17 | 18 | 19 | 20      | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28      | 29 | 30 | 31      |  |
| R     | 0  |    |    |    | MUXSEL5 |    | 0  |    |    |    |    |    | MUXSEL1 |    |    | MUXSEL0 |  |
| W     |    |    |    |    |         |    |    |    |    |    |    |    |         |    |    |         |  |
| Reset | 0  | 0  | 0  | 0  | 0       | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0       | 0  | 0  | 0       |  |

### SIU\_IMUX1 field descriptions

| Field             | Description                                                                                                                                                                                                                                                                                                                                                   |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–19<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                       |
| 20–21<br>MUXSEL5  | Selects the source pad for FlexCAN D receive : CNRXD.<br><br>00 Selects input from "CNRXD_MCANRX1_GPIO247" pad (PCR: 247).<br>01 Selects input from "EMIOS15_IRQ1_CNRXD_RCH1_C_GPIO194" pad (PCR: 194).<br>10 Reserved<br>11 Reserved                                                                                                                         |
| 22–27<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                       |
| 28–29<br>MUXSEL1  | Selects the source pad for MCAN 1 Rx data : MCANRX1.<br><br>00 Selects input from "ETPUC15_MCANRX1_RCH2_C_GPIO456" pad (PCR: 456).<br>01 Selects input from "CNRXD_MCANRX1_GPIO247" pad (PCR: 247).<br>10 Selects input from "EMIOS17_ETPUB1_MCANRX1_RCH3_C_GPIO196" pad (PCR: 196).<br>11 Reserved                                                           |
| 30–31<br>MUXSEL0  | Selects the source pad for MCAN 0 Rx data : MCANRX0.<br><br>00 Selects input from "ETPUA19_PCSD4_MCANRX0_GPIO133" pad (PCR: 133).<br>01 Selects input from "CNRXA_RXDA_MCANRX0_GPIO84" pad (PCR: 84).<br>10 Selects input from "TXDB_PCSD1_MCANRX0_GPIO91" pad (PCR: 91).<br>11 Selects input from "PCSC5_MCANRX0_SOUT_D_LVDSM_PCSD0_GPIO243" pad (PCR: 243). |

## 8.2.68 Input Multiplexing Register2 (SIU\_IMUX2)

The SIU\_IMUX registers specify the source of peripheral inputs that may come from multiple package pins. The registers allow selection from 4 sources for each input. Each SIU\_IMUX register contains 16 inputs, each with 2 bits (4 sources).

Some registers are reserved for future use, and some fields within some of the registers are also reserved.

Address: 0h base + 1208h offset = 1208h

|       |   |   |   |   |   |   |   |   |          |   |          |    |         |    |         |    |
|-------|---|---|---|---|---|---|---|---|----------|---|----------|----|---------|----|---------|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8        | 9 | 10       | 11 | 12      | 13 | 14      | 15 |
| R     | 0 |   |   |   |   |   |   |   | MUXSEL11 |   | MUXSEL10 |    | MUXSEL9 |    | MUXSEL8 |    |
| W     |   |   |   |   |   |   |   |   |          |   |          |    |         |    |         |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0        | 0 | 0        | 0  | 0       | 0  | 0       | 0  |

|       |         |    |         |    |         |    |    |    |    |    |    |    |    |    |    |    |
|-------|---------|----|---------|----|---------|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 16      | 17 | 18      | 19 | 20      | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | MUXSEL7 |    | MUXSEL6 |    | MUXSEL5 |    | 0  |    |    |    |    |    |    |    |    |    |
| W     |         |    |         |    |         |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0       | 0  | 0       | 0  | 0       | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**SIU\_IMUX2 field descriptions**

| Field             | Description                                                                                                                                                                                                                                                                                                        |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–7<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                            |
| 8–9<br>MUXSEL11   | Selects the source pad for DSPI D peripheral chip select 0 : PCSD0.<br>00 Selects input from "PCSB1_PCSD0_GPIO106" pad (PCR: 106).<br>01 Selects input from "ETPUC28_PCSD0_RCH6_C_MAB2_GPIO469" pad (PCR: 469).<br>10 Selects input from "PCSC5_MCANRX0_SOUT_D_LVDSM_PCSD0_GPIO243" pad (PCR: 243).<br>11 Reserved |
| 10–11<br>MUXSEL10 | Selects the source pad for DSPI D data input : SIND.<br>00 Selects input from "PCSC3_SCK_D_LVDSM_SIND_GPIO241" pad (PCR: 241).<br>01 Selects input from "ETPUC31_SIND_RCH9_C_GPIO472" pad (PCR: 472).<br>10 Selects input from "EMIOS11_SIND_GPIO190" pad (PCR: 190).<br>11 Reserved                               |
| 12–13<br>MUXSEL9  | Selects the source pad for DSPI D clock : SCKD.<br>00 Selects input from "PCSC2_SCK_D_LVDSP_SCKD_GPIO240" pad (PCR: 240).<br>01 Selects input from "ETPUC29_SCKD_RCH7_C_GPIO470" pad (PCR: 470).<br>10 Selects input from "EMIOS10_SCKD_GPIO189" pad (PCR: 189).<br>11 Reserved                                    |
| 14–15<br>MUXSEL8  | Selects the source pad for DSPI C peripheral chip select 0 : PCSC0.<br>00 Selects input from "PCSC0_SOUT_C_LVDSM_GPIO238" pad (PCR: 238).<br>01 Selects input from "PCSB5_PCSC0_SENT5_B_FEC_RXD2_GPIO110" pad (PCR: 110).<br>10 Selects input from "EMIOS28_PCSC0_GPIO434" pad (PCR: 434).<br>11 Reserved          |
| 16–17<br>MUXSEL7  | Selects the source pad for DSPI C data input : SINC.<br>00 Selects input from "SINC_SCK_C_LVDSM_GPIO236" pad (PCR: 236).<br>01 Selects input from "PCSB3_SINC_SENT3_B_GPIO108" pad (PCR: 108).<br>10 Reserved<br>11 Reserved                                                                                       |
| 18–19<br>MUXSEL6  | Selects the source pad for DSPI C clock : SCKC.<br>00 Selects input from "SCKC_SCK_C_LVDSP_GPIO235" pad (PCR: 235).<br>01 Selects input from "PCSB4_SCKC_SENT4_B_FEC_MDC_GPIO109" pad (PCR: 109).<br>10 Reserved<br>11 Reserved                                                                                    |

Table continues on the next page...

## SIU\_IMUX2 field descriptions (continued)

| Field             | Description                                                                                                                                                                                                                                        |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 20–21<br>MUXSEL5  | <p>Selects the source pad for DSPI B peripheral chip select 0 : PCSB0.</p> <p>00 Selects input from "PCSB0_PCSD2_GPIO105" pad (PCR: 105).<br/> 01 Selects input from "EMIOS24_PCSB0_GPIO203" pad (PCR: 203).<br/> 10 Reserved<br/> 11 Reserved</p> |
| 22–31<br>Reserved | <p>This field is reserved.<br/> This read-only field is reserved and always has the value 0.</p>                                                                                                                                                   |

## 8.2.69 Input Multiplexing Register3 (SIU\_IMUX3)

The SIU\_IMUX registers specify the source of peripheral inputs that may come from multiple package pins. The registers allow selection from 4 sources for each input. Each SIU\_IMUX register contains 16 inputs, each with 2 bits (4 sources).

Some registers are reserved for future use, and some fields within some of the registers are also reserved

Address: 0h base + 120Ch offset = 120Ch

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |    |         |    |         |    |         |    |         |    |         |    |         |    |         |    |
|-------|----|----|---------|----|---------|----|---------|----|---------|----|---------|----|---------|----|---------|----|
| Bit   | 16 | 17 | 18      | 19 | 20      | 21 | 22      | 23 | 24      | 25 | 26      | 27 | 28      | 29 | 30      | 31 |
| R     | 0  |    | MUXSEL6 |    | MUXSEL5 |    | MUXSEL4 |    | MUXSEL3 |    | MUXSEL2 |    | MUXSEL1 |    | MUXSEL0 |    |
| W     |    |    |         |    |         |    |         |    |         |    |         |    |         |    |         |    |
| Reset | 0  | 0  | 0       | 0  | 0       | 0  | 0       | 0  | 0       | 0  | 0       | 0  | 0       | 0  | 0       | 0  |

## SIU\_IMUX3 field descriptions

| Field            | Description                                                                                                                                                                                                                                         |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–17<br>Reserved | <p>This field is reserved.<br/> This read-only field is reserved and always has the value 0.</p>                                                                                                                                                    |
| 18–19<br>MUXSEL6 | <p>Selects the source pad for External interrupt request 6 : IRQ6.</p> <p>00 Selects input from "TCRCLKB_IRQ6_GPIO146" pad (PCR: 146).<br/> 01 Selects input from "TCRCLKC_IRQ6_ANB40_GPI440" pad (PCR: 440).<br/> 10 Reserved<br/> 11 Reserved</p> |
| 20–21<br>MUXSEL5 | <p>Selects the source pad for External interrupt request 5 : IRQ5.</p> <p>00 Selects input from "PLLCFG1_IRQ5_SOUTD_SIPI_TXP_GPIO209" pad (PCR: 209).<br/> 01 Selects input from "ETPUC14_IRQ5_SDIN0_PSI5A_RCH1_C_GPIO455" pad (PCR: 455).</p>      |

Table continues on the next page...

**SIU\_IMUX3 field descriptions (continued)**

| Field            | Description                                                                                                                                                                                                                                                           |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                  | 10 Reserved<br>11 Reserved                                                                                                                                                                                                                                            |
| 22–23<br>MUXSEL4 | Selects the source pad for External interrupt request 4 : IRQ4.<br><br>00 Selects input from "PLLCFG0_IRQ4_ETPUC9_SIP1_TXN_GPIO208" pad (PCR: 208).<br>01 Selects input from "ETPUC13_IRQ4_SDOUT0_PSI5A_RCH0_C_GPIO454" pad (PCR: 454).<br>10 Reserved<br>11 Reserved |
| 24–25<br>MUXSEL3 | Selects the source pad for External interrupt request 3 : IRQ3.<br><br>00 Selects input from "BOOTCFG1_IRQ3_ETPUC0_SIP1_RXP_GPIO212" pad (PCR: 212).<br>01 Selects input from "ETPUC12_IRQ3_MCANTX1_GPIO453" pad (PCR: 453).<br>10 Reserved<br>11 Reserved            |
| 26–27<br>MUXSEL2 | Selects the source pad for External interrupt request 2 : IRQ2.<br><br>00 Selects input from "BOOTCFG0_IRQ2_GPIO211" pad (PCR: 211).<br>01 Selects input from "ETPUC11_IRQ2_GPIO452" pad (PCR: 452).<br>10 Reserved<br>11 Reserved                                    |
| 28–29<br>MUXSEL1 | Selects the source pad for External interrupt request 1 : IRQ1.<br><br>00 Selects input from "EMIOS15_IRQ1_CNFXD_RCH1_C_GPIO194" pad (PCR: 194).<br>01 Selects input from "ETPUC10_IRQ1_SENT5_B_GPIO451" pad (PCR: 451).<br>10 Reserved<br>11 Reserved                |
| 30–31<br>MUXSEL0 | Selects the source pad for External interrupt request 0 : IRQ0.<br><br>00 Selects input from "EMIOS14_IRQ0_CNFXD_RCH0_C_GPIO193" pad (PCR: 193).<br>01 Selects input from "ETPUC9_IRQ0_SENT4_B_ANB44_GPI450" pad (PCR: 450).<br>10 Reserved<br>11 Reserved            |

**8.2.70 Input Multiplexing Register4 (SIU\_IMUX4)**

The SIU\_IMUX registers specify the source of peripheral inputs that may come from multiple package pins. The registers allow selection from 4 sources for each input. Each SIU\_IMUX register contains 16 inputs, each with 2 bits (4 sources).

Some registers are reserved for future use, and some fields within some of the registers are also reserved

## Memory Map and Register Definition

Address: 0h base + 1210h offset = 1210h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |         |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|---------|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13      | 14 | 15 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    | MUXSEL8 |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |         |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0       | 0  | 0  |

|       |    |    |    |    |    |    |    |    |         |    |         |    |         |    |         |    |
|-------|----|----|----|----|----|----|----|----|---------|----|---------|----|---------|----|---------|----|
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24      | 25 | 26      | 27 | 28      | 29 | 30      | 31 |
| R     | 0  |    |    |    |    |    |    |    | MUXSEL3 |    | MUXSEL2 |    | MUXSEL1 |    | MUXSEL0 |    |
| W     |    |    |    |    |    |    |    |    |         |    |         |    |         |    |         |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0       | 0  | 0       | 0  | 0       | 0  | 0       | 0  |

### SIU\_IMUX4 field descriptions

| Field             | Description                                                                                                                                                                                                                                                                  |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–13<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                      |
| 14–15<br>MUXSEL8  | Selects the source pad for eSCI E transmit : TXDE.<br><br>00 Selects input from "PCSA4_SCKE_TXDE_FEC_RXCLK_GPIO100" pad (PCR: 100).<br>01 Selects input from "MDO3_TXDE_GPIO223" pad (PCR: 223).<br>10 Reserved<br>11 Reserved                                               |
| 16–23<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                      |
| 24–25<br>MUXSEL3  | Selects the source pad for eSCI B transmit : TXDB.<br><br>00 Selects input from "TXDB_PCSD1_MCANRX0_GPIO91" pad (PCR: 91).<br>01 Selects input from "ETPUC21_TXDB_GPIO462" pad (PCR: 462).<br>10 Reserved<br>11 Reserved                                                     |
| 26–27<br>MUXSEL2  | Selects the source pad for eSCI B receive : RXDB.<br><br>00 Selects input from "RXDB_PCSD5_GPIO92" pad (PCR: 92).<br>01 Selects input from "ETPUC22_RXDB_GPIO463" pad (PCR: 463).<br>10 Reserved<br>11 Reserved                                                              |
| 28–29<br>MUXSEL1  | Selects the source pad for eSCI A receive : RXDA.<br><br>00 Selects input from "RXDA_GPIO90" pad (PCR: 90).<br>01 Selects input from "CNRXA_RXDA_MCANRX0_GPIO84" pad (PCR: 84).<br>10 Selects input from "ETPUC20_RXDA_SDIN0_PSI5B_GPIO461" pad (PCR: 461).<br>11 Reserved   |
| 30–31<br>MUXSEL0  | Selects the source pad for eSCI A transmit : TXDA.<br><br>00 Selects input from "TXDA_GPIO89" pad (PCR: 89).<br>01 Selects input from "CNTXA_TXDA_MCANTX0_GPIO83" pad (PCR: 83).<br>10 Selects input from "ETPUC19_TXDA_SDOUT0_PSI5B_GPIO460" pad (PCR: 460).<br>11 Reserved |



### 8.2.71 Input Multiplexing Register5 (SIU\_IMUX5)

The SIU\_IMUX registers specify the source of peripheral inputs that may come from multiple package pins. The registers allow selection from 4 sources for each input. Each SIU\_IMUX register contains 16 inputs, each with 2 bits (4 sources).

Some registers are reserved for future use, and some fields within some of the registers are also reserved

Address: 0h base + 1214h offset = 1214h

|       |          |   |          |   |          |   |          |   |          |   |          |    |         |    |         |    |
|-------|----------|---|----------|---|----------|---|----------|---|----------|---|----------|----|---------|----|---------|----|
| Bit   | 0        | 1 | 2        | 3 | 4        | 5 | 6        | 7 | 8        | 9 | 10       | 11 | 12      | 13 | 14      | 15 |
| R     | MUXSEL15 |   | MUXSEL14 |   | MUXSEL13 |   | MUXSEL12 |   | MUXSEL11 |   | MUXSEL10 |    | MUXSEL9 |    | MUXSEL8 |    |
| W     |          |   |          |   |          |   |          |   |          |   |          |    |         |    |         |    |
| Reset | 0        | 0 | 0        | 0 | 0        | 0 | 0        | 0 | 0        | 0 | 0        | 0  | 0       | 0  | 0       | 0  |

|       |         |    |         |    |         |    |         |    |         |    |         |    |         |    |         |    |
|-------|---------|----|---------|----|---------|----|---------|----|---------|----|---------|----|---------|----|---------|----|
| Bit   | 16      | 17 | 18      | 19 | 20      | 21 | 22      | 23 | 24      | 25 | 26      | 27 | 28      | 29 | 30      | 31 |
| R     | MUXSEL7 |    | MUXSEL6 |    | MUXSEL5 |    | MUXSEL4 |    | MUXSEL3 |    | MUXSEL2 |    | MUXSEL1 |    | MUXSEL0 |    |
| W     |         |    |         |    |         |    |         |    |         |    |         |    |         |    |         |    |
| Reset | 0       | 0  | 0       | 0  | 0       | 0  | 0       | 0  | 0       | 0  | 0       | 0  | 0       | 0  | 0       | 0  |

#### SIU\_IMUX5 field descriptions

| Field           | Description                                                                                                                                                                                                                                                                    |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–1<br>MUXSEL15 | <p>Selects the source pad for eMIOS 1 channel 23 : EMIOS31.</p> <p>00 Selects input from "EMIOS31_PCSC5_GPIO437" pad (PCR: 437).<br/>           01 Selects input from "ANB23_ETPUC8_EMIOS31_GPI491" pad (PCR: 491).<br/>           10 Reserved<br/>           11 Reserved</p>  |
| 2–3<br>MUXSEL14 | <p>Selects the source pad for eMIOS 1 channel 22 : EMIOS30.</p> <p>00 Selects input from "EMIOS30_PCSC2_GPIO436" pad (PCR: 436).<br/>           01 Selects input from "ANB22_ETPUC9_EMIOS30_GPI490" pad (PCR: 490).<br/>           10 Reserved<br/>           11 Reserved</p>  |
| 4–5<br>MUXSEL13 | <p>Selects the source pad for eMIOS 1 channel 21 : EMIOS29.</p> <p>00 Selects input from "EMIOS29_PCSC1_GPIO435" pad (PCR: 435).<br/>           01 Selects input from "ANB21_ETPUC10_EMIOS29_GPI489" pad (PCR: 489).<br/>           10 Reserved<br/>           11 Reserved</p> |
| 6–7<br>MUXSEL12 | <p>Selects the source pad for eMIOS 1 channel 20 : EMIOS28.</p> <p>00 Selects input from "EMIOS28_PCSC0_GPIO434" pad (PCR: 434).<br/>           01 Selects input from "ANB20_ETPUC11_EMIOS28_GPI488" pad (PCR: 488).<br/>           10 Reserved<br/>           11 Reserved</p> |
| 8–9<br>MUXSEL11 | <p>Selects the source pad for eMIOS 1 channel 19 : EMIOS27.</p> <p>00 Selects input from "EMIOS27_PCSB3_GPIO433" pad (PCR: 433).</p>                                                                                                                                           |

*Table continues on the next page...*

## SIU\_IMUX5 field descriptions (continued)

| Field             | Description                                                                                                                                                                                                                          |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | 01 Selects input from "ANB19_ETPUC12_EMIO27_GPI487" pad (PCR: 487).<br>10 Reserved<br>11 Reserved                                                                                                                                    |
| 10–11<br>MUXSEL10 | Selects the source pad for eMIOS 1 channel 18 : EMIO26.<br>00 Selects input from "EMIO26_PCSB2_GPIO432" pad (PCR: 432).<br>01 Selects input from "ANB18_ETPUC13_EMIO26_GPI486" pad (PCR: 486).<br>10 Reserved<br>11 Reserved         |
| 12–13<br>MUXSEL9  | Selects the source pad for eMIOS 1 channel 17 : EMIO25.<br>00 Selects input from "EMIO25_PCSB1_GPIO204" pad (PCR: 204).<br>01 Selects input from "ANB17_ETPUC14_EMIO25_GPI485" pad (PCR: 485).<br>10 Reserved<br>11 Reserved         |
| 14–15<br>MUXSEL8  | Selects the source pad for eMIOS 1 channel 16 : EMIO24.<br>00 Selects input from "EMIO24_PCSB0_GPIO203" pad (PCR: 203).<br>01 Selects input from "ANB16_ETPUC15_EMIO24_GPI484" pad (PCR: 484).<br>10 Reserved<br>11 Reserved         |
| 16–17<br>MUXSEL7  | Selects the source pad for eMIOS 0 channel 23 : EMIO23.<br>00 Selects input from "EMIO23_ETPUB7_RCH9_C_GPIO202" pad (PCR: 202).<br>01 Selects input from "ANB15_ETPUA24_EMIO23_GPI483" pad (PCR: 483).<br>10 Reserved<br>11 Reserved |
| 18–19<br>MUXSEL6  | Selects the source pad for eMIOS 0 channel 22 : EMIO22.<br>00 Selects input from "EMIO22_ETPUB6_RCH8_C_GPIO201" pad (PCR: 201).<br>01 Selects input from "ANB14_ETPUA25_EMIO22_GPI482" pad (PCR: 482).<br>10 Reserved<br>11 Reserved |
| 20–21<br>MUXSEL5  | Selects the source pad for eMIOS 0 channel 21 : EMIO21.<br>00 Selects input from "EMIO21_ETPUB5_RCH7_C_GPIO200" pad (PCR: 200).<br>01 Selects input from "ANB13_ETPUA26_EMIO21_GPI481" pad (PCR: 481).<br>10 Reserved<br>11 Reserved |
| 22–23<br>MUXSEL4  | Selects the source pad for eMIOS 0 channel 20 : EMIO20.<br>00 Selects input from "EMIO20_ETPUB4_RCH6_C_GPIO199" pad (PCR: 199).<br>01 Selects input from "ANB12_ETPUA27_EMIO20_GPI480" pad (PCR: 480).<br>10 Reserved<br>11 Reserved |
| 24–25<br>MUXSEL3  | Selects the source pad for eMIOS 0 channel 19 : EMIO19.<br>00 Selects input from "EMIO19_ETPUB3_RCH5_C_GPIO198" pad (PCR: 198).                                                                                                      |

Table continues on the next page...

**SIU\_IMUX5 field descriptions (continued)**

| Field            | Description                                                                                                                                                                                                                                     |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                  | 01 Selects input from "ANB11_ETPUA28_EMIO19_GPI479" pad (PCR: 479).<br>10 Reserved<br>11 Reserved                                                                                                                                               |
| 26–27<br>MUXSEL2 | Selects the source pad for eMIOS 0 channel 18 : EMIO18.<br><br>00 Selects input from "EMIO18_ETPUB2_RCH4_C_GPIO197" pad (PCR: 197).<br>01 Selects input from "ANB10_ETPUA29_EMIO18_GPI478" pad (PCR: 478).<br>10 Reserved<br>11 Reserved        |
| 28–29<br>MUXSEL1 | Selects the source pad for eMIOS 0 channel 17 : EMIO17.<br><br>00 Selects input from "EMIO17_ETPUB1_MCANRX1_RCH3_C_GPIO196" pad (PCR: 196).<br>01 Selects input from "ANB9_ETPUA30_EMIO17_GPI477" pad (PCR: 477).<br>10 Reserved<br>11 Reserved |
| 30–31<br>MUXSEL0 | Selects the source pad for eMIOS 0 channel 16 : EMIO16.<br><br>00 Selects input from "EMIO16_ETPUB0_MCANTX1_RCH2_C_GPIO195" pad (PCR: 195).<br>01 Selects input from "ANB8_ETPUA31_EMIO16_GPI476" pad (PCR: 476).<br>10 Reserved<br>11 Reserved |

**8.2.72 Input Multiplexing Register7 (SIU\_IMUX7)**

The SIU\_IMUX registers specify the source of peripheral inputs that may come from multiple package pins. The registers allow selection from 4 sources for each input. Each SIU\_IMUX register contains 16 inputs, each with 2 bits (4 sources).

Some registers are reserved for future use, and some fields within some of the registers are also reserved.

Address: 0h base + 121Ch offset = 121Ch

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
| R     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

## SIU\_IMUX7 field descriptions

| Field             | Description                                                                                                                                                                                                                                            |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–1<br>MUXSEL15   | <p>Selects the source pad for eTPU A channel 31 : ETPUA31.</p> <p>00 Selects input from "ETPUA31_PCSC4_RCH9_A_GPIO145" pad (PCR: 145).<br/> 01 Selects input from "ANB8_ETPUA31_EMIO16_GPIO476" pad (PCR: 476).<br/> 10 Reserved<br/> 11 Reserved</p>  |
| 2–3<br>MUXSEL14   | <p>Selects the source pad for eTPU A channel 30 : ETPUA30.</p> <p>00 Selects input from "ETPUA30_PCSC3_RCH8_A_GPIO144" pad (PCR: 144).<br/> 01 Selects input from "ANB9_ETPUA30_EMIO17_GPIO477" pad (PCR: 477).<br/> 10 Reserved<br/> 11 Reserved</p>  |
| 4–5<br>MUXSEL13   | <p>Selects the source pad for eTPU A channel 29 : ETPUA29.</p> <p>00 Selects input from "ETPUA29_PCSC2_RCH7_A_GPIO143" pad (PCR: 143).<br/> 01 Selects input from "ANB10_ETPUA29_EMIO18_GPIO478" pad (PCR: 478).<br/> 10 Reserved<br/> 11 Reserved</p> |
| 6–7<br>MUXSEL12   | <p>Selects the source pad for eTPU A channel 28 : ETPUA28.</p> <p>00 Selects input from "ETPUA28_PCSC1_RCH6_A_GPIO142" pad (PCR: 142).<br/> 01 Selects input from "ANB11_ETPUA28_EMIO19_GPIO479" pad (PCR: 479).<br/> 10 Reserved<br/> 11 Reserved</p> |
| 8–9<br>MUXSEL11   | <p>Selects the source pad for eTPU A channel 27 : ETPUA27.</p> <p>00 Selects input from "ETPUA27_IRQ15_RCH5_A_GPIO141" pad (PCR: 141).<br/> 01 Selects input from "ANB12_ETPUA27_EMIO20_GPIO480" pad (PCR: 480).<br/> 10 Reserved<br/> 11 Reserved</p> |
| 10–11<br>MUXSEL10 | <p>Selects the source pad for eTPU A channel 26 : ETPUA26.</p> <p>00 Selects input from "ETPUA26_IRQ14_RCH4_A_GPIO140" pad (PCR: 140).<br/> 01 Selects input from "ANB13_ETPUA26_EMIO21_GPIO481" pad (PCR: 481).<br/> 10 Reserved<br/> 11 Reserved</p> |
| 12–13<br>MUXSEL9  | <p>Selects the source pad for eTPU A channel 25 : ETPUA25.</p> <p>00 Selects input from "ETPUA25_IRQ13_RCH3_A_GPIO139" pad (PCR: 139).<br/> 01 Selects input from "ANB14_ETPUA25_EMIO22_GPIO482" pad (PCR: 482).<br/> 10 Reserved<br/> 11 Reserved</p> |
| 14–15<br>MUXSEL8  | <p>Selects the source pad for eTPU A channel 24 : ETPUA24.</p> <p>00 Selects input from "ETPUA24_IRQ12_RCH2_A_GPIO138" pad (PCR: 138).<br/> 01 Selects input from "ANB15_ETPUA24_EMIO23_GPIO483" pad (PCR: 483).<br/> 10 Reserved<br/> 11 Reserved</p> |

Table continues on the next page...

**SIU\_IMUX7 field descriptions (continued)**

| Field             | Description                                                                             |
|-------------------|-----------------------------------------------------------------------------------------|
| 16–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |

**8.2.73 Input Multiplexing Register10 (SIU\_IMUX10)**

The SIU\_IMUX registers specify the source of peripheral inputs that may come from multiple package pins. The registers allow selection from 4 sources for each input. Each SIU\_IMUX register contains 16 inputs, each with 2 bits (4 sources).

Some registers are reserved for future use, and some fields within some of the registers are also reserved.

Address: 0h base + 1228h offset = 1228h

|       |          |   |          |   |          |   |          |   |          |   |          |    |         |    |         |    |
|-------|----------|---|----------|---|----------|---|----------|---|----------|---|----------|----|---------|----|---------|----|
| Bit   | 0        | 1 | 2        | 3 | 4        | 5 | 6        | 7 | 8        | 9 | 10       | 11 | 12      | 13 | 14      | 15 |
| R     |          |   |          |   |          |   |          |   |          |   |          |    |         |    |         |    |
| W     |          |   |          |   |          |   |          |   |          |   |          |    |         |    |         |    |
|       | MUXSEL15 |   | MUXSEL14 |   | MUXSEL13 |   | MUXSEL12 |   | MUXSEL11 |   | MUXSEL10 |    | MUXSEL9 |    | MUXSEL8 |    |
| Reset | 0        | 0 | 0        | 0 | 0        | 0 | 0        | 0 | 0        | 0 | 0        | 0  | 0       | 0  | 0       | 0  |

|       |    |    |    |    |    |    |    |    |         |    |          |    |         |    |         |    |
|-------|----|----|----|----|----|----|----|----|---------|----|----------|----|---------|----|---------|----|
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24      | 25 | 26       | 27 | 28      | 29 | 30      | 31 |
| R     |    |    |    |    |    |    |    |    |         |    |          |    |         |    |         |    |
| W     |    |    |    |    |    |    |    |    |         |    |          |    |         |    |         |    |
|       |    |    |    |    | 0  |    |    |    | MUXSEL3 |    | Reserved |    | MUXSEL1 |    | MUXSEL0 |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0       | 0  | 0        | 0  | 0       | 0  | 0       | 0  |

**SIU\_IMUX10 field descriptions**

| Field           | Description                                                                                                                                                                                                                                          |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–1<br>MUXSEL15 | Selects the source pad for eTPU C channel 15 : ETPUC15.<br><br>00 Selects input from "ETPUC15_MCANRX1_RCH2_C_GPIO456" pad (PCR: 456).<br>01 Selects input from "ANB16_ETPUC15_EMIO24_GPI484" pad (PCR: 484).<br>10 Reserved<br>11 Reserved           |
| 2–3<br>MUXSEL14 | Selects the source pad for eTPU C channel 14 : ETPUC14.<br><br>00 Selects input from "ETPUC14_IRQ5_SDIN0_PSI5A_RCH1_C_GPIO455" pad (PCR: 455).<br>01 Selects input from "ANB17_ETPUC14_EMIO25_GPI485" pad (PCR: 485).<br>10 Reserved<br>11 Reserved  |
| 4–5<br>MUXSEL13 | Selects the source pad for eTPU C channel 13 : ETPUC13.<br><br>00 Selects input from "ETPUC13_IRQ4_SDOUT0_PSI5A_RCH0_C_GPIO454" pad (PCR: 454).<br>01 Selects input from "ANB18_ETPUC13_EMIO26_GPI486" pad (PCR: 486).<br>10 Reserved<br>11 Reserved |
| 6–7<br>MUXSEL12 | Selects the source pad for eTPU C channel 12 : ETPUC12.                                                                                                                                                                                              |

Table continues on the next page...

## SIU\_IMUX10 field descriptions (continued)

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | 00 Selects input from "ETPUC12_IRQ3_MCANTX1_GPIO453" pad (PCR: 453).<br>01 Selects input from "ANB19_ETPUC12_EMIO27_GPIO487" pad (PCR: 487).<br>10 Reserved<br>11 Reserved                                                                                                                                                                                                                                                                                 |
| 8–9<br>MUXSEL11   | Selects the source pad for eTPU C channel 11 : ETPUC11.<br>00 Selects input from "ETPUC11_IRQ2_GPIO452" pad (PCR: 452).<br>01 Selects input from "ANB20_ETPUC11_EMIO28_GPIO488" pad (PCR: 488).<br>10 Reserved<br>11 Reserved                                                                                                                                                                                                                              |
| 10–11<br>MUXSEL10 | Selects the source pad for eTPU C channel 10 : ETPUC10.<br>00 Selects input from "ETPUC10_IRQ1_SENT5_B_GPIO451" pad (PCR: 451).<br>01 Selects input from "ANB21_ETPUC10_EMIO29_GPIO489" pad (PCR: 489).<br>10 Reserved<br>11 Reserved                                                                                                                                                                                                                      |
| 12–13<br>MUXSEL9  | Selects the source pad for eTPU C channel 9 : ETPUC9.<br>00 Selects input from "ETPUC9_IRQ0_SENT4_B_ANB44_GPIO450" pad (PCR: 450).<br>01 Selects input from "ANB22_ETPUC9_EMIO30_GPIO490" pad (PCR: 490).<br>10 Selects input from "PLLCFG0_IRQ4_ETPUC9_SIPI_TXN_GPIO208" pad (PCR: 208).<br>11 Reserved                                                                                                                                                   |
| 14–15<br>MUXSEL8  | Selects the source pad for eTPU C channel 8 : ETPUC8.<br>00 Selects input from "ETPUC8_PCSE5_SENT3_B_GPIO449" pad (PCR: 449).<br>01 Selects input from "ANB23_ETPUC8_EMIO31_GPIO491" pad (PCR: 491).<br>10 Reserved<br>11 Reserved                                                                                                                                                                                                                         |
| 16–23<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                    |
| 24–25<br>MUXSEL3  | Selects the source pad for eTPU C channel 3 : ETPUC3.<br>00 Selects input from "ETPUC3_SENT4_A_ANB45_GPIO444" pad (PCR: 444).<br>01 Selects input from "PLLCFG2_ETPUC3_PCSE1_SIPI_REFCLK_GPIO210" pad (PCR: 210).<br><br><b>NOTE:</b> The PLLCFG2 function is only effective during reset. After reset, the pin takes other alternate functions, and these other functions are the ones intended to be selected by this mux.<br>10 Reserved<br>11 Reserved |
| 26–27<br>Reserved | This field is reserved.                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 28–29<br>MUXSEL1  | Selects the source pad for eTPU C channel 1 : ETPUC1.<br>00 Selects input from "ETPUC1_SENT1_A_ANB42_GPIO442" pad (PCR: 442).<br>01 Selects input from "WKPCFG_NMI_ETPUC1_SIPI_RXN_GPIO213" pad (PCR: 213).<br><br><b>NOTE:</b> The WKPCFG function is only effective during reset. After reset, the pin takes other alternate functions, and these other functions are the ones intended to be selected by this mux.                                      |

Table continues on the next page...

**SIU\_IMUX10 field descriptions (continued)**

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                  | 10 Reserved<br>11 Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 30–31<br>MUXSEL0 | Selects the source pad for eTPU C channel 0 : ETPUC0.<br><br>00 Selects input from "ETPUC0_SENT0_A_ANB41_GPI441" pad (PCR: 441).<br>01 Selects input from "BOOTCFG1_IRQ3_ETPUC0_SIPI_RXP_GPIO212" pad (PCR: 212).<br><br><b>NOTE:</b> The BOOTCFG1 function is only effective during reset. After reset, the pin takes other alternate functions, and these other functions are the ones intended to be selected by this mux.<br><br>10 Reserved<br>11 Reserved |

**8.2.74 Input Multiplexing Register12 (SIU\_IMUX12)**

The SIU\_IMUX registers specify the source of peripheral inputs that may come from multiple package pins. The registers allow selection from 4 sources for each input. Each SIU\_IMUX register contains 16 inputs, each with 2 bits (4 sources).

Some registers are reserved for future use, and some fields within some of the registers are also reserved.

Address: 0h base + 1230h offset = 1230h

|       |    |    |    |    |    |    |    |    |    |    |    |    |         |    |    |    |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|---------|----|----|----|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12      | 13 | 14 | 15 |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |         |    |    |    |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |         |    |    |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0       | 0  | 0  | 0  |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28      | 29 | 30 | 31 |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    | MUXSEL2 |    |    |    |
| W     |    |    |    |    |    |    |    |    |    |    |    |    | 0       |    |    |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0       | 0  | 0  | 0  |

**SIU\_IMUX12 field descriptions**

| Field             | Description                                                                                                                                                                                                                             |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–25<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                 |
| 26–27<br>MUXSEL2  | Selects the source pad for eTPU C TCR clock : TCRCLKC.<br><br>00 Selects input from "TCRCLKC_IRQ6_ANB40_GPI440" pad (PCR: 440).<br>01 Selects input from "ETPUC4_SENT5_A_TCRCLKC_GPIO445" pad (PCR: 445).<br>10 Reserved<br>11 Reserved |
| 28–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                 |

## SIU\_IMUX12 field descriptions (continued)

| Field | Description |
|-------|-------------|
|-------|-------------|

## 8.3 SIU\_B

SIU\_B contains a set of mirrored PCR configuration registers, GPO data registers, and GPI data registers for GPIO[240:243] balls.

By default after reset, the mirrored registers are disabled and provide no function. The disabled state is indicated by PA[2:0] = 111b in the mirrored PCR (PCR<sub>nM</sub>) register.

When a mirrored PCR<sub>nM</sub> is enabled by writing a different value to its PA[2:0]:

- The PCR<sub>nM</sub> has priority over the equivalent non-mirrored PCR register.
- The PCR<sub>nM</sub> controls the ball function and characteristics defined by its implemented bit fields.
- The mirrored GPO data (GPDO<sub>nM</sub>) and GPI data (GPDIn<sub>M</sub>) registers control and reflect the logic level.

When the mirrored registers are enabled, writes to the non-mirrored registers have no effect.

Offsets for SIU\_B registers are relative to SIU\_B's base address, which differs from SIU's base address.

### SIU\_B memory map

| Address offset (hex) | Register name                                           | Width (in bits) | Access | Reset value                 | Section/ page             |
|----------------------|---------------------------------------------------------|-----------------|--------|-----------------------------|---------------------------|
| 220                  | Pad Configuration Register Mirrored (SIU_B_PCR240M)     | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.3.1/389</a> |
| 222                  | Pad Configuration Register Mirrored (SIU_B_PCR241M)     | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.3.1/389</a> |
| 224                  | Pad Configuration Register Mirrored (SIU_B_PCR242M)     | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.3.1/389</a> |
| 226                  | Pad Configuration Register Mirrored (SIU_B_PCR243M)     | 16              | R/W    | <a href="#">See section</a> | <a href="#">8.3.1/389</a> |
| 6F0                  | GPIO Pin Data Output Register Mirrored (SIU_B_GPDO240M) | 8               | R/W    | 00h                         | <a href="#">8.3.2/392</a> |
| 6F1                  | GPIO Pin Data Output Register Mirrored (SIU_B_GPDO241M) | 8               | R/W    | 00h                         | <a href="#">8.3.2/392</a> |
| 6F2                  | GPIO Pin Data Output Register Mirrored (SIU_B_GPDO242M) | 8               | R/W    | 00h                         | <a href="#">8.3.2/392</a> |

*Table continues on the next page...*



**SIU\_B memory map (continued)**

| Address offset (hex) | Register name                                           | Width (in bits) | Access | Reset value | Section/ page             |
|----------------------|---------------------------------------------------------|-----------------|--------|-------------|---------------------------|
| 6F3                  | GPIO Pin Data Output Register Mirrored (SIU_B_GPDO243M) | 8               | R/W    | 00h         | <a href="#">8.3.2/392</a> |
| 8F0                  | GPIO Pin Data Input Register Mirrored (SIU_B_GPDI240M)  | 8               | R      | 00h         | <a href="#">8.3.3/393</a> |
| 8F1                  | GPIO Pin Data Input Register Mirrored (SIU_B_GPDI241M)  | 8               | R      | 00h         | <a href="#">8.3.3/393</a> |
| 8F2                  | GPIO Pin Data Input Register Mirrored (SIU_B_GPDI242M)  | 8               | R      | 00h         | <a href="#">8.3.3/393</a> |
| 8F3                  | GPIO Pin Data Input Register Mirrored (SIU_B_GPDI243M)  | 8               | R      | 00h         | <a href="#">8.3.3/393</a> |

**8.3.1 Pad Configuration Register Mirrored (SIU\_B\_PCRnM)**

By default after reset, each mirrored register is disabled and provides no function. The disabled state is indicated by the reset value of PA[2:0] = 111b.

When a mirrored PCR<sub>n</sub>M is enabled by writing a different value to its PA[2:0]:

- The PCR<sub>n</sub>M has priority over the equivalent non-mirrored PCR register.
- The PCR<sub>n</sub>M controls the ball function and characteristics defined by its implemented bit fields.

When the mirrored registers are enabled, writes to the non-mirrored registers have no effect.

These 16-bit registers are read from or written to as:

- 16-bit values aligned on 16-bit boundaries, or
- 32-bit values aligned on 32-bit address boundaries.

The PCRs allow software to control the following range of electrical characteristics of the external pads:

- Weak pullup/down enable/disable
- Weak pullup/down selection
- Slew-rate selection for outputs
- Drive strength selection for outputs
- Input buffer enable (when direction is configured for output)
- Input hysteresis enable/disable
- Open drain/push-pull output selection
- Multiplexed function selection
- Data direction selection

**NOTE**

The actual characteristics available in any SIU\_B\_PCR $n$ M depend on the type of pad it controls. See the SIU\_B\_PCR $n$ M definition table.

All device pin names begin with the primary function, followed by the alternate function, and then GPIO. See [External signal descriptions, multiplexing, and attributes](#) for how to identify the source module for primary and alternate functions and the description of these signals. For an additional signal description, see the chapter for the module that uses the signal.

There are a number of input signals on the device that have multiple external pins as input sources. Only one input source can be active at any time for these pins. To achieve this, a set of multiplexers controlled by the SIU\_IMUX registers select which input will drive the function. See [Input Multiplexing Register0 \(SIU\\_IMUX0\)](#) to [Input Multiplexing Register12 \(SIU\\_IMUX12\)](#) for more details.

[Table 8-2027](#) defines the symbols for PCR $n$ M fields' reset values that appear in [Table 8-2028](#).

**Table 8-2030. SIU\_B\_PCR $n$ M field reset-value symbols**

| Symbol | Meaning                                                                                                              |
|--------|----------------------------------------------------------------------------------------------------------------------|
| —      | Not implemented in the PCR $n$ M. Software should write a default value of 0.                                        |
| 0 or 1 | Function is implemented. For the exact definition of the 0 or 1 reset value, see the applicable field's description. |

**Table 8-2031. SIU\_B\_PCR[240:243]M reset values**

| PCR <i>n</i> M | Offset<br>(hex) | PCR <i>n</i> M field reset values |     |     |     |     |      |      |     |     |      |      |     |     |
|----------------|-----------------|-----------------------------------|-----|-----|-----|-----|------|------|-----|-----|------|------|-----|-----|
|                |                 | PA2                               | PA1 | PA0 | OBE | IBE | DSC1 | DSC0 | ODE | HYS | SRC1 | SRC0 | WPE | WPS |
| PCR240M        | 220             | 1                                 | 1   | 1   | 0   | 0   | —    | —    | 0   | 0   | 0    | 0    | 1   | 1   |
| PCR241M        | 222             | 1                                 | 1   | 1   | 0   | 0   | —    | —    | 0   | 0   | 0    | 0    | 1   | 1   |
| PCR242M        | 224             | 1                                 | 1   | 1   | 0   | 0   | —    | —    | 0   | 0   | 0    | 0    | 1   | 1   |
| PCR243M        | 226             | 1                                 | 1   | 1   | 0   | 0   | —    | —    | 0   | 0   | 0    | 0    | 1   | 1   |

Address: 0h base + 220h offset + (2d × i), where i=0d to 3d

| Bit   | 0 | 1 | 2 | 3  | 4 | 5 | 6   | 7   | 8   | 9 | 10  | 11  | 12  | 13 | 14  | 15  |
|-------|---|---|---|----|---|---|-----|-----|-----|---|-----|-----|-----|----|-----|-----|
| Read  | 0 |   |   | PA |   |   | OBE | IBE | DSC |   | ODE | HYS | SRC |    | WPE | WPS |
| Write |   |   |   |    |   |   |     |     |     |   |     |     |     |    |     |     |
| Reset | 0 | 0 | 0 | 1  | 1 | 1 | 0   | 0   | *   | * | 0   | 0   | 0   | 0  | 1   | 1   |

\* Notes:

- DSC field: This function is not implemented. Software should write a default value of 0 to each bit position.

## SIU\_B\_PCRnM field descriptions

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–2<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 3–5<br>PA       | Pin assignment. Selects the function of a multiplexed pad. A separate port enable output signal from the SIU is asserted for each value of this register. The size of the field can be from 1 to 3 bits, depending on the amount of multiplexing on the pad.<br><br>000 GPIO.<br>001 Primary function. See the description of the corresponding SIU_PCRn.<br>010 Alternate 1. See the description of the corresponding SIU_PCRn.<br>011 Alternate 2. See the description of the corresponding SIU_PCRn.<br>100 Alternate 3. See the description of the corresponding SIU_PCRn.<br>101 Invalid value.<br>110 Invalid value.<br>111 SIU_B's mirrored PCRnM, GPDO <sub>n</sub> M, and GPDInM registers are disabled and provide no function. |
| 6<br>OBE        | Output buffer enable. Enables the pad as an output and drives the output buffer enable signal.<br><br>0 Disable output buffer for the pad.<br>1 Enable output buffer for the pad.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 7<br>IBE        | Input buffer enable. Enables the pad as an input and drives the input buffer enable signal.<br><br>0 Disable input buffer for the pad.<br>1 Enable input buffer for the pad.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 8–9<br>DSC      | Drive strength control. Controls the pad drive strength.<br><br>00 10 pF drive strength.<br>01 20 pF drive strength.<br>10 30 pF drive strength.<br>11 50 pF drive strength.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 10<br>ODE       | Open drain output enable. Controls output driver configuration for the pads. Either open drain or push/pull driver configurations can be selected. This feature applies to output pins only.<br><br>0 Disable open drain for the pad (push/pull driver enabled).<br>1 Enable open drain for the pad.                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 11<br>HYS       | Input hysteresis. Controls whether hysteresis is enabled for the pad.<br><br>0 Disable hysteresis for the pad.<br>1 Enable hysteresis for the pad.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 12–13<br>SRC    | Slew rate control. Controls slew rate for the pad. Slew rate control pertains to pins with slow or medium I/O pad types, and the output signals are driven according to the value of this field. Actual slew rate depends on the pad type and load. Refer to the electrical specifications for this information.<br><br>00 Minimum slew rate.<br>01 Medium slew rate.<br>10 Invalid value.<br>11 Maximum slew rate.                                                                                                                                                                                                                                                                                                                       |
| 14<br>WPE       | Weak pull up/down enable. Controls whether the weak pullup/down devices are enabled/disabled for the pad. Pullup/down devices are enabled by default.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

*Table continues on the next page...*

**SIU\_B\_PCRnM field descriptions (continued)**

| Field     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|           | 0 Disable weak pull device for the pad.<br>1 Enable weak pull device for the pad.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 15<br>WPS | Weak pull up/down select. Controls whether weak pullup or weak pulldown devices are used for the pad when weak pullup/down devices are enabled. During reset, the weak pull up/down behavior is controlled by the WKPCFG bit of the Miscellaneous DCF client, or by the WKPCFG pin when operating in legacy mode ((LEG=1 in the Miscellaneous DCF client). The WPS bit determines whether weak pullup or pulldown devices are used after reset, or for pads in which the WKPCFG pin does not determine the reset weak pullup/down state.<br><br>0 Pulldown is enabled for the pad.<br>1 Pullup is enabled for the pad. |

**8.3.2 GPIO Pin Data Output Register Mirrored (SIU\_B\_GPDO<sub>n</sub>M)**

By default after reset, each mirrored register is disabled and provides no function. The disabled state is indicated by PA[2:0] = 111b in the mirrored PCR (PCR<sub>n</sub>M) register.

When a mirrored PCR<sub>n</sub>M is enabled by writing a different value to its PA[2:0], the mirrored GPO data (GPDO<sub>n</sub>M) and GPI data (GPDInM) registers control and reflect the logic level.

When the mirrored registers are enabled, writes to the non-mirrored registers have no effect.

Each GPDO register specifies the output data for the function assigned to the GPIO[*n*] pin. The *n* notation in the SIU\_B\_GPDO<sub>n</sub>M register name relate to the [*n*] in GPIO[*n*] signal name. For example, SIU\_B\_GPDO240M contains the PDO240 bit for CNTXD\_GPIO240.

Software writes to the SIU\_B\_GPDO<sub>n</sub>M registers to drive data out on the external pin. Each register drives one external pin, which allows independent control of the pin. Writes to the SIU\_B\_GPDO<sub>n</sub>M register are not reflected to the pin when a primary or alternate function is assigned or when an input function is assigned to the pin.

If the direction of a GPIO signal changes from input to output, the SIU\_B\_GPDO<sub>n</sub>M register value is automatically driven out to the external pin without the need for a software update.

Writes to a SIU\_B\_GPDO<sub>n</sub>M register have no effect when a primary or alternate function is assigned.

Address: 0h base + 6F0h offset + (1d × i), where i=0d to 3d

|       |   |   |   |   |   |   |                  |   |
|-------|---|---|---|---|---|---|------------------|---|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6                | 7 |
| Read  | 0 |   |   |   |   |   | PDO <sub>n</sub> |   |
| Write |   |   |   |   |   |   |                  |   |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0                | 0 |

### SIU\_B\_GPDO<sub>n</sub>M field descriptions

| Field                 | Description                                                                                                                                                                                                                                                                                                                                 |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–6<br>Reserved       | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                     |
| 7<br>PDO <sub>n</sub> | Pin Data Out. This bit stores the data to be driven out on the external GPIO. If the register is read, it returns the value written.<br><br>0 Logic low value is driven to the external GPIO pin when the pin is configured as an output.<br>1 Logic high value is driven to the external GPIO pin when the pin is configured as an output. |

### 8.3.3 GPIO Pin Data Input Register Mirrored (SIU\_B\_GPDI<sub>n</sub>M)

By default after reset, each mirrored register is disabled and provides no function. The disabled state is indicated by PA[2:0] = 111b in the mirrored PCR (PCR<sub>n</sub>M) register.

When a mirrored PCR<sub>n</sub>M is enabled by writing a different value to its PA[2:0], the mirrored GPO data (GPDO<sub>n</sub>M) and GPI data (GPDI<sub>n</sub>M) registers control and reflect the logic level.

When the mirrored registers are enabled, writes to the non-mirrored registers have no effect.

Each SIU\_B\_GPDI<sub>n</sub>M register specifies the input state for the GPDI[*n*] pin. The *n* notation in the SIU\_GPDI<sub>n</sub> register name relates to the [*n*] in GPIO[*n*] signal name. For example, SIU\_B\_GPDI240M contains the PDI240 bit for CNTXD\_GPIO240.

Software reads the SIU\_B\_GPDI<sub>n</sub>M registers to get the input state of the external GPIO pin. Each GPDI register contains the input state of one external GPIO pin. If a GPDI register is configured as output, and the input buffer enable bit is set to 1 in the PCR<sub>n</sub>M register, the SIU\_B\_GPDI<sub>n</sub>M register reflects the state of the output pin.

Address: 0h base + 8F0h offset + (1d × i), where i=0d to 3d

|       |   |   |   |   |   |   |                  |   |
|-------|---|---|---|---|---|---|------------------|---|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6                | 7 |
| Read  | 0 |   |   |   |   |   | PDI <sub>n</sub> |   |
| Write |   |   |   |   |   |   |                  |   |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0                | 0 |

**SIU\_B\_GPDInM field descriptions**

| Field           | Description                                                                                                                                                         |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–6<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                             |
| 7<br>PDIn       | Pin Data In. This bit reflects the state on the external GPIO/GPI pin. If PCRnM[IBE] is 1, then:<br>0 Signal on pin is logic zero.<br>1 Signal on pin is logic one. |

## 8.4 Functional Description

The following sections provide a functional overview of the SIU operation.

### 8.4.1 Pad configuration

The pad configuration registers (SIU\_PCR) in the SIU allow software control over the following electrical characteristics of the external pads:

- Weak pullup/down enable/disable
- Weak pullup/down selection
- Slew-rate selection for outputs
- Drive strength selection for outputs
- Input buffer enable (when direction is configured for output)
- Input hysteresis enable/disable
- Open drain/push-pull output selection
- Multiplexed function selection
- Data direction selection

The pad configuration registers are provided to allow centralized control over external pins that are shared by more than one module. Each pad configuration register controls a single pin.

### 8.4.2 Interrupts

There are 16 external interrupt inputs  $\overline{\text{IRQ}}[0]–\overline{\text{IRQ}}[15]$  to the SIU. The IRQ inputs can be configured for rising-edge events, falling-edge events, or both.

External interrupt requests are triggered by rising- and/or falling-edge events that are enabled by setting a bit in:

- IRQ rising-edge event enable register (SIU\_IREER)
- IRQ falling-edge event enable register (SIU\_IFEER)

If the bit is set in both registers, both rising- and falling-edge events trigger an interrupt request. Each IRQ has a counter that tracks the number of system clock cycles that occur between the rising- and falling-edge events. An IRQ counter exists for each IRQ rising- or falling-edge event enable bit.

The digital filter length field in the IRQ digital filter register (SIU\_IDFR) specifies the minimum number of system clocks that the IRQ signal must hold a logic value to qualify the edge-triggered event as a valid state change. When the number of system clocks in the IRQ counter equals the value in the digital filter length field, the IRQ state latches and the IRQ counter is cleared.

If the previous filtered state of the IRQ does not match the current state, and the rising- or falling-edge event is enabled, the IRQ flag bit is set to 1. For example, the IRQ flag bit is set if a rising-edge event occurs under the following conditions:

- Previous filtered IRQ state was a logic 0
- Current latched IRQ state is a logic 1
- Rising-edge event is enabled for the IRQ

When the counter for an IRQ is not enabled, the state of the IRQ is held in the current and previous state latches. The IRQ counter operates independently of the IRQ or overrun flag bit. Clearing the IRQ flag or overrun flag bits does not clear or reload the counter.

### 8.4.2.1 External interrupts

The IRQ signals map to 16 independent interrupt requests output from the SIU. The IRQ flag bit is set when a rising-edge and/or falling-edge event occurs for the IRQ. An external IRQ signal is asserted when all of the following occur:

- Enable bit is set in the IRQ rising- and/or falling-edge event registers (SIU\_IREER, SIU\_IFEER)
- IRQ flag bit is set in the external interrupt status register (SIU\_EISR)
- Enable bit is set in the DMA/Interrupt request enable register (SIU\_DIRER)
- Select bit is cleared in the DMA/Interrupt select register (SIU\_DIRSR)

The NMI pin function or platform SWT can generate either an NMI or a critical interrupt. FCCU may also generate NMI if any fault is configured to cause NMI and this fault is triggered.

When WKPCFG\_NMI\_GPIO213 is enabled as NMI, the pin will override the PCR configuration after reset. SIU\_DIRER selects between critical and non maskable interrupt use, SIU\_EISR reports status of NMI and SIU\_IFEER selects edge sensitivity of NMI input. See in the figure below the NMI and critical interrupt connections.

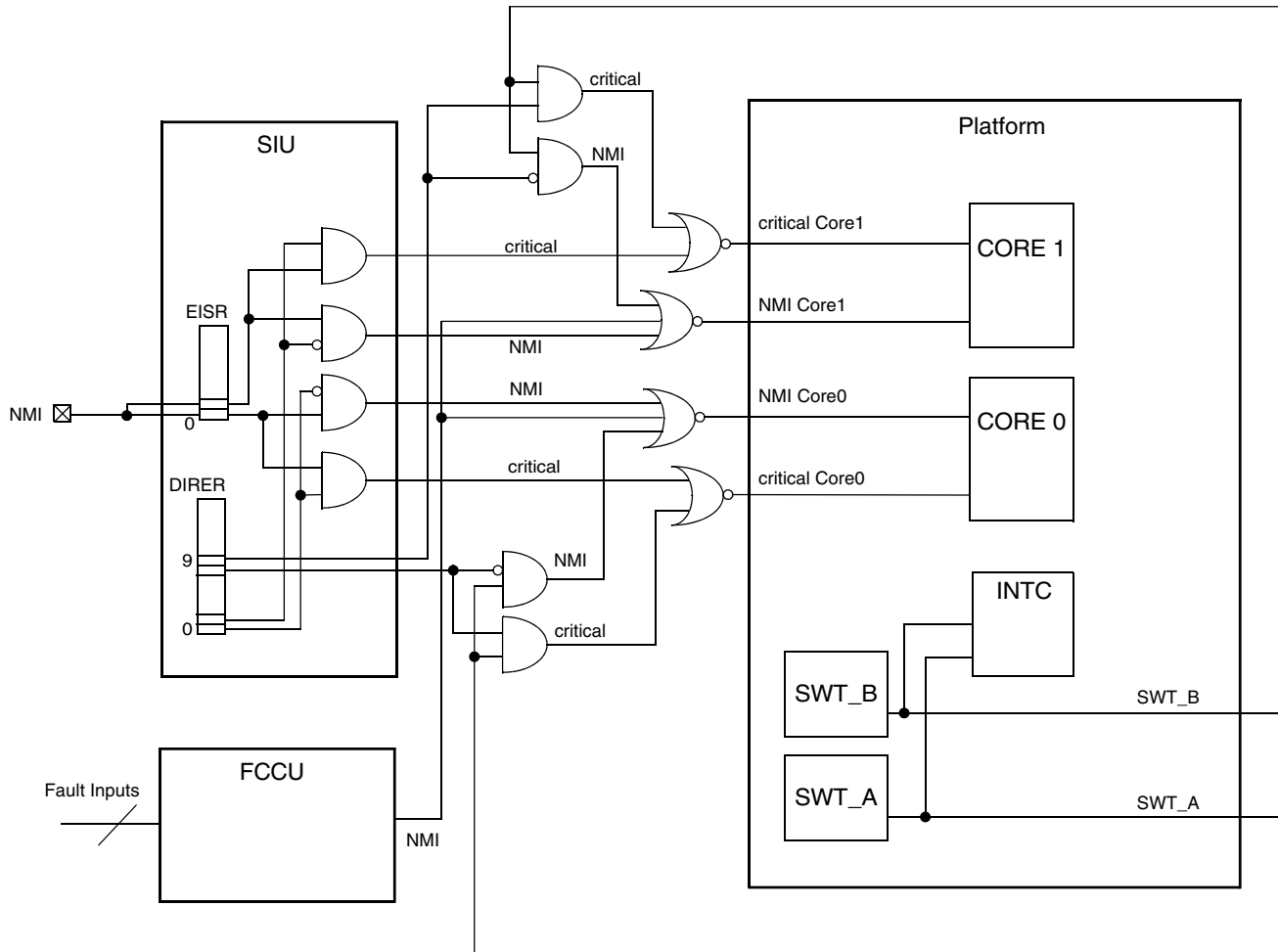


Figure 8-1605. NMI, critical, and SWT interrupts

### 8.4.2.2 DMA transfers

DMA IRQ signals ( $\overline{\text{IRQ}}[0]$  through  $\overline{\text{IRQ}}[3]$ ) map to four independent DMA transfer or interrupt request outputs configured in the SIU. A DMA transfer or interrupt request asserts when all of the following occur:

- IRQ flag bit is set in the external interrupt status register (SIU\_EISR)
- Enable bit is set in the DMA transfer or interrupt request enable register (SIU\_DIRER)
- Select bit is set in the DMA transfer or interrupt request select register (SIU\_DIRSR)

The SIU receives a “DMA transfer done” signal for each DMA or interrupt request transmitted. When the “DMA done” signal asserts, the IRQ flag bit is cleared.



### 8.4.2.3 Overruns

An overrun IRQ exists for each overrun flag bit in the overrun status register (SIU\_OSR). An overrun IRQ asserts when all of the following occur:

- Enable bit is set in the IRQ rising- and/or falling-edge event registers (SIU\_IREER, SIU\_IFEER)
- IRQ flag bit is set in the external interrupt status register (SIU\_EISR)
- Bit is set in the overrun request enable and overrun status registers (SIU\_ORER, SIU\_OSR)
- Rising- or falling-edge event triggers an interrupt request

The SIU outputs one overrun IRQ bit that is the logical OR of all of the IRQ overrun bits.

### 8.4.2.4 Edge-detect events

An IRQ asserts when:

- An edge-detect event is enabled
- An edge-detect event occurs

To assert an IRQ when an edge-detect event occurs:

1. Set the enable bit in the IRQ rising- and falling-edge event enable registers (SIU\_IREER, SIU\_IFEER)
2. Set the enable bits for the DMA/Interrupt request enable register (SIU\_DIRER)

The IRQ bit is set in the external IRQ status register (SIU\_EISR) when an edge-detect event occurs for that IRQ. The following figure illustrates the behavior of DMA/interrupt requests within the SIU.

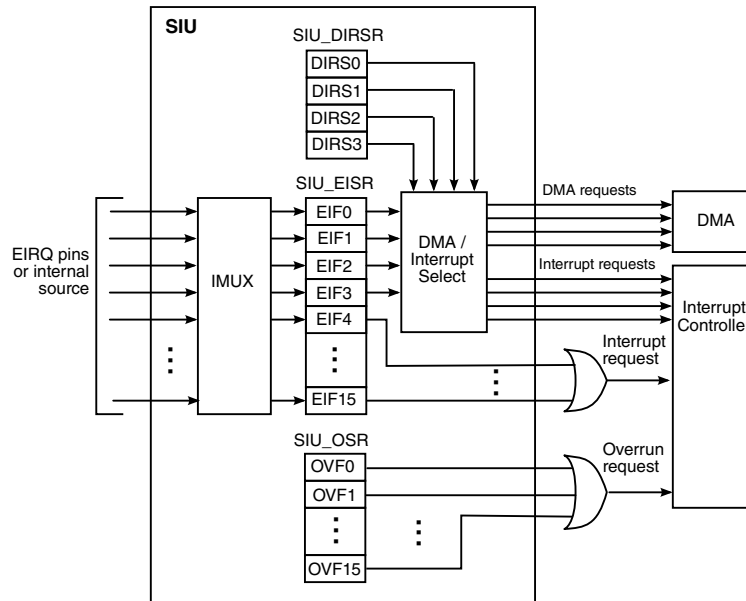


Figure 8-1606. SIU DMA/interrupt request diagram

### 8.4.3 General Purpose I/O (GPIO)

All GPIO functions for the device are provided by the SIU. Each device pad that has a GPIO signal has a pin configuration register (PCR) in the SIU where the GPIO function is selected. In addition, each device GPIO signal has an input data register (SIU\_GPDI $n$ ) and an output data register (SIU\_GPDO $n$ ).

The SIU also implements several parallel GPIO registers (SIU\_PGPDO $n$  and SIU\_PGPDIn) that can be used to access up to 32 GPIO bits in a single- and word-sized accesses. The values read/written to these parallel registers are coherent with the data read/written to the SIU\_GPDO $n$  and SIU\_GPDI $n$  registers.

### 8.4.4 Pin serialization for DSPI

The SIU has four 32-bit outputs to DSPI modules. Each data frame contains the result of the selection, bit by bit, of three sources of data. Each data frame could receive data from eTPU, eMIOS, or an internally implemented GPO register. These registers are defined to provide compliance with the AutoSAR requirements for a PORT driver and support for coherent bit manipulation of up to 16 serialized data bits in a multitasking or multi-processor system. Because the DSPI frame size has been extended to a maximum of 32 bits, the coherent bit manipulation mechanism is partitioned into two separate 16-bit halves of the frame. The purpose of these registers is to allow any combination of bits in each half of the 32-bit serialized data frame to be updated with a single 32-bit write

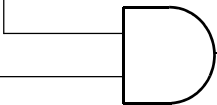
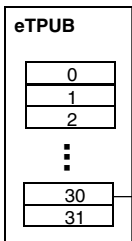
operation, while allowing other bits to maintain their previous state. This is accomplished by writing a 16-bit masked value coherently with an update value contained in a 16 bit output field, and only updating those bits in the output register for which the corresponding mask bit is set.

The selection of which bits are actually output on the DSPI data frame is determined by a separate signal selection register. Each register bit enables a path from a single eTPU channel, eMIOS channel, and masked GPO register bit to the equivalent bit position in the DSPI channel frame. Note that each DSPI bit position can be driven by multiple inputs. Software must ensure that the correct bit sources are enabled for correct insertion of the required bit in the bit frame. Multiple sources are logically ORed, which provides the potential for combining outputs from multiple timer channels and data registers to produce more complex bit behavior. An example of how these registers are used to create the DSPI serialized output is shown below.

Functional Description

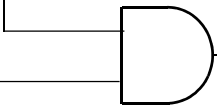
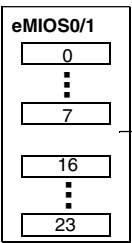
eTPU Select Register (for DSPI\_A)

|        | 0  | 1  | 2  | 3  | 4  | 5  | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|--------|----|----|----|----|----|----|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Chan # | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5  | 4  | 3  | 2  | 1  | 0  | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| Enable | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |



eMIOS Select Register (for DSPI\_A)

|        | 0   | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | 10  | 11  | 12  | 13  | 14  | 15  | 16   | 17   | 18   | 19   | 20   | 21   | 22   | 23   | 24  | 25  | 26  | 27  | 28  | 29  | 30  | 31  |
|--------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|------|------|------|------|------|-----|-----|-----|-----|-----|-----|-----|-----|
| Chan # | 0_7 | 0_6 | 0_5 | 0_4 | 0_3 | 0_2 | 0_1 | 0_0 | 1_0 | 1_1 | 1_2 | 1_3 | 1_4 | 1_5 | 1_6 | 1_7 | 0_16 | 0_17 | 0_18 | 0_19 | 0_20 | 0_21 | 0_22 | 0_23 | 0_0 | 0_1 | 0_2 | 0_3 | 0_4 | 0_5 | 0_6 | 0_7 |
| Enable | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |

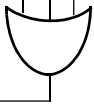
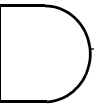
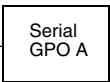


DSPIH/L Select Register (for DSPI\_A)

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

SIU\_DSPIL[MASK17]

SIU\_DSPIL[DATA17]



DSPIA Output Register

| Frame bit                    | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17               | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|------------------------------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Function & Channel Selection |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    | eTPUB Channel 30 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

Figure 8-1607. Example of pin multiplexing

## 8.4.5 Internal multiplexing

The internal multiplexing select registers (SIU\_ISEL4–8, SIU\_EISR, and SIU\_DISR) select the input sources for the following components:

- eTPUA[24:29] inputs
- eQADC command FIFO trigger sources
- SIU external interrupt request signals

### 8.4.5.1 eQADC external trigger input multiplexing

The eQADC external trigger inputs can connect to one of the following pins:

- External pin
- eTPU channel
- eMIOS channel
- Periodic interrupt timer

The input source for each eQADC external trigger is configured in the eQADC Command FIFO Trigger Source Select Registers (SIU\_ISEL4–7). For example, you can select one of the following signals to source the ETRIG[0] input trigger for the eQADC:

- TXDC\_ETRIG[0]\_GPIO[244] pin
- eTPUA[30] channel output
- eMIOS\_1 channel 2 output
- Extended trigger source selection

All ETRIG inputs are multiplexed in the same manner. If an eTPU or eMIOS channel is selected as an ETRIG input to the eQADC, you can activate the alternate function of that eTPU or eMIOS signal on the external pin.

## 8.4.6 Trigger Burst Generators (TBG)

The Trigger Burst Generator module generates triggers for the eQADC CFIFOs based on a trigger window provided by several sources connected to the IMUX. eTPU, eMIOS, PIT and external SoC inputs can be selected by the IMUX, thus allowed to provide a time window in which the triggers are generated by the Trigger Burst Generator (TBG) module. The TBG is in bypass out of reset, thus backwards compatible functionality is provided. In this backwards compatible setting, the triggers are generated by the signals connected to the IMUX, thus TBG is transparent to the trigger generation. The following figure shows the Trigger Burst Generator block diagram and the waveforms associated with the input trigger window, the output trigger burst (on the left) and the counter waveform (on the right).

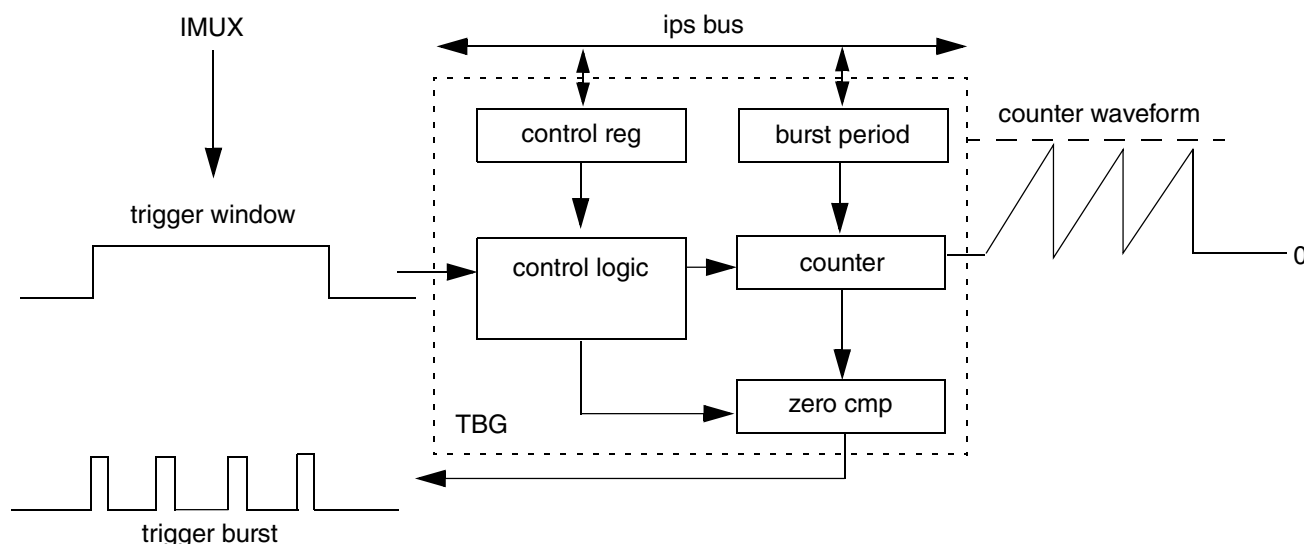


**Figure 8-1608. Trigger Burst Generator block diagram**

The following figure is a block diagram of the Trigger Burst Generator module interconnection with other modules. The TBG is connected between the multiplexers controlled by ISEL4–7 and eQADC\_A/b CIFIFO trigger inputs. There are six CFIFOs on each eQADC and two instances of the eQADC modules, thus 12 TBG instances are used in order to provide trigger burst for all CFIFOs.

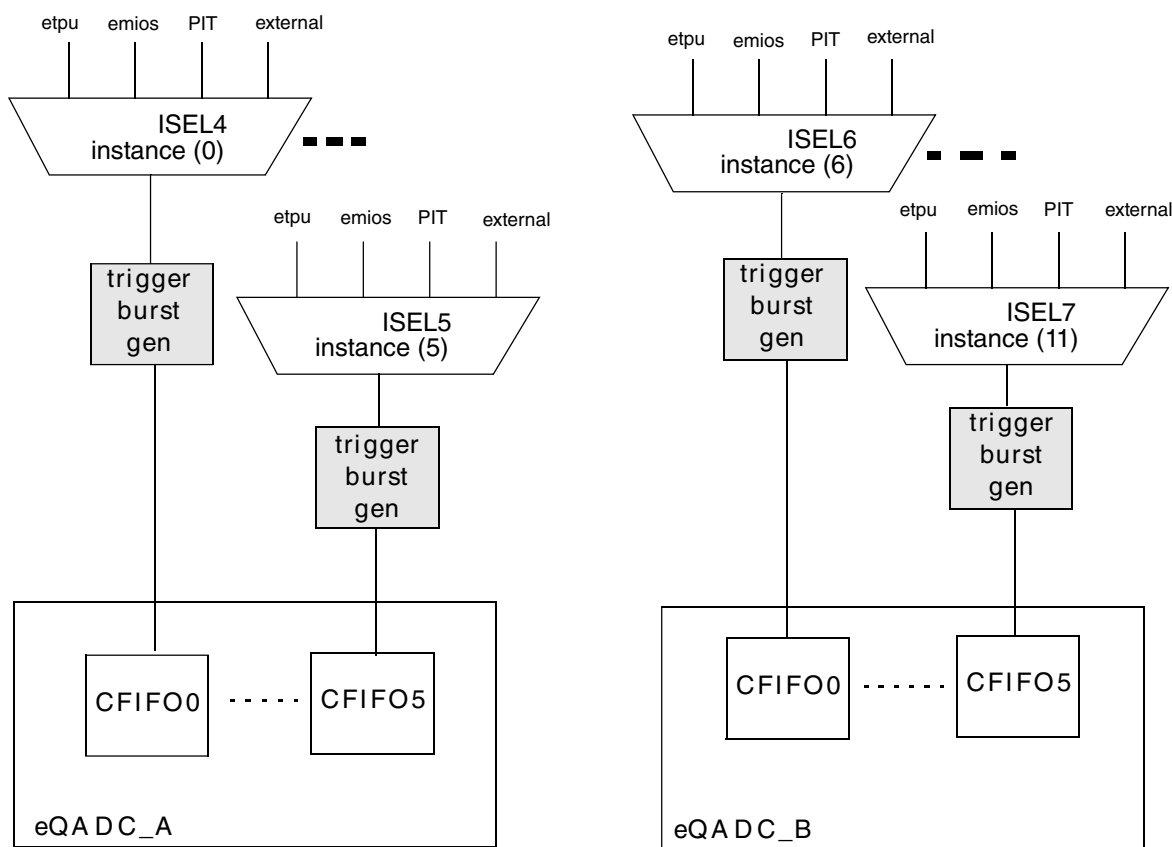


**Figure 8-1609. Trigger Burst Generator interconnection**

The TBG instances generate a continuous sequence of triggers while the signals selected by ISEL4–7 are asserted. When the trigger sources deassert, then the trigger burst is stopped. The output is identified as the trigger window in the subsequent sections. The opening of a trigger window means that the ISEL4–7 mux outputs enable the TBGs to start generating triggers.

#### 8.4.6.1 Clock selection for the trigger burst

There are three clocks that can be selected for the TBG module. The system clock, the ADC0 clock, and the ADC1 clock. If no clock is selected the output trigger remains negated. It is necessary to have GEN = 0 when changing the CLKSEL to avoid unwanted triggers.

Figure 8-1610 shows the timing diagram of the signals when the peripheral clock (per\_clk) is selected to generate the trigger burst. In this case, the trigger counter is enabled to count in all per\_clk pulses. Therefore, the time interval between two trigger pulses is calculated as (TRIGPER + 1) per\_clk periods. The figure shows the counter values when TRIGPER is configured to 4.

The first trigger occurs after one clock cycle after the trigger window begins. And the last trigger pulse occurs before the trigger window ends. Therefore, there is no trigger pulse outside the trigger window.

NOTE

When the trigger period register is zero (TRIGPER = 0), the trigger output switches to ON one per\_clk period after the trigger window is opened, and remains ON (no transition) until the trigger window is closed.

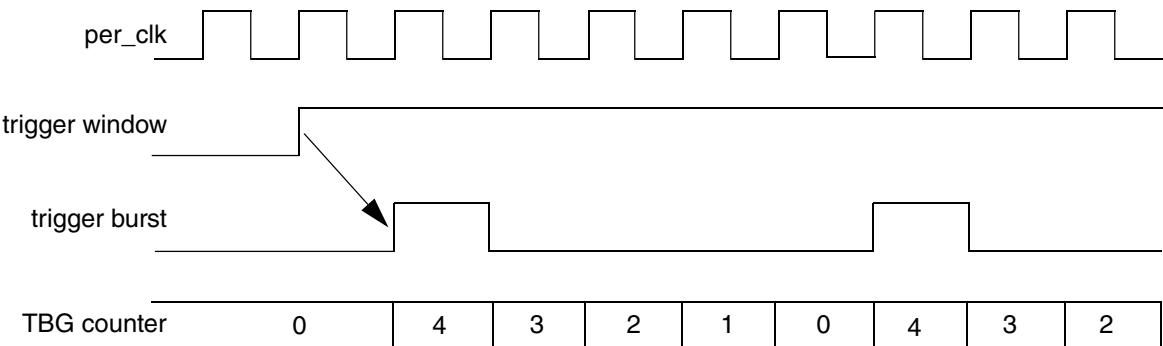


Figure 8-1610. Trigger Burst Generator timing diagram when per\_clk is selected

Figure 8-1611 shows the timing diagram of the signals when the ADC0/1 clocks are selected to generate the trigger burst. In this case, the trigger counter is enabled to count only in the positive edge events of that ADC clock. Therefore, the time interval between two trigger pulses is calculated as (TRIGPER + 1) ADC clock periods.

NOTE

When the trigger period register is zero (TRIGPER = 0) the trigger output pulse occurs for each ADC clock pulse.

The first trigger occurs one system clock cycle after the first detected rising edge of the selected ADC clock after the opening of the trigger window. And the last trigger pulse occurs before the trigger window closes. Therefore, there is no trigger pulse outside the trigger window.



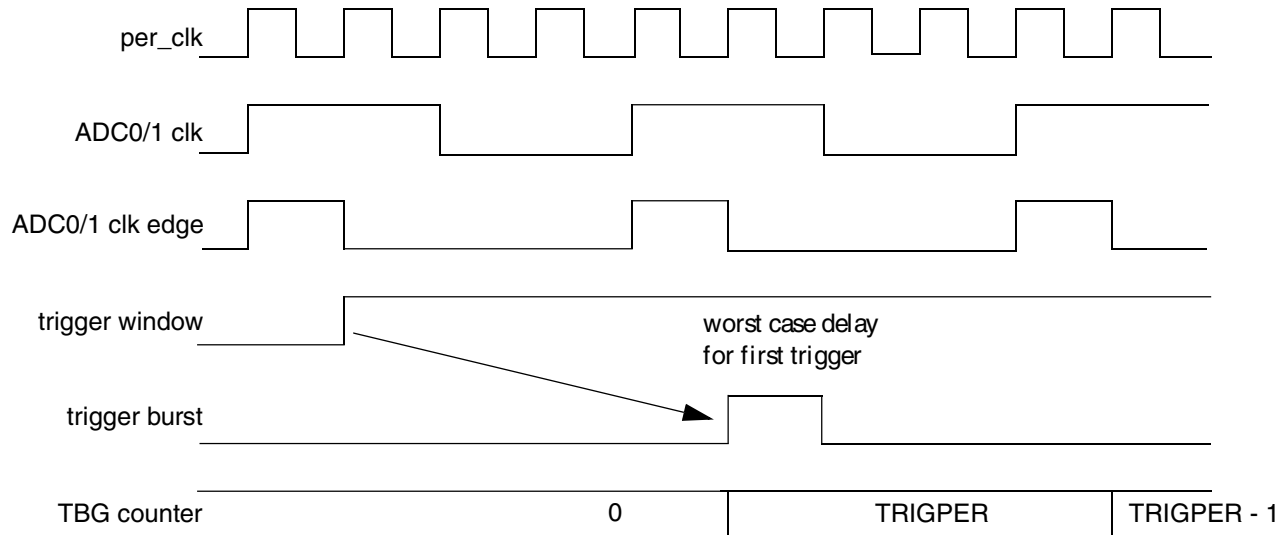


**Figure 8-1611. Trigger Burst Generator timing diagram when ADC0/1 is selected**

#### 8.4.6.2 Trigger generator configuration procedure

The trigger burst generator (TBG) is disabled out of reset because the GEN bit at the SIU\_TBGR\_CR\_A/Bn register from reset state is clear. This means the trigger window and internal/external selector input signals are bypassed to the respective output ports.

It is recommended to disable the TBG by clearing the GEN bit when the trigger period has to be changed. However, the input trigger window signal at the input port is bypassed to the trigger burst output. If this unwanted trigger must be avoided, the GEN bit must be maintained set and the CLKSEL bit must be configured to all 0 (the trigger counter is cleared at this time). Then it is possible to safely change the trigger period.



# Chapter 9

## Crossbar Switch (XBAR)

### 9.1 Chip-specific XBAR information

This chip has one instance of the XBAR module.

#### 9.1.1 XBAR master and slave assignments

The following table lists the XBAR physical port numbers and logical IDs for all master ports on this SoC.

- Each port number matches the default priority assigned to the corresponding physical master port. This default priority equals the reset value of the priority field for each master port in the PRS*n* registers.
- A priority value of 0 is the highest priority. There is no "disabled" value for the priority.
- A Nexus\_3 module and core data bus share the same physical master port for each core.

The logical master ID corresponds to the logical address provided by the master module and is unique for each module. The logical master IDs are used by the bus masters connected to the XBAR. The Nexus master is identified by setting the MSB in the 4-bit field that supplies the master ID number.

**Table 9-1. XBAR master ports and logical master IDs**

| Module            | Physical master port | Logical master ID | Comment                                              |
|-------------------|----------------------|-------------------|------------------------------------------------------|
| Core0 instruction | 0                    | 0                 |                                                      |
| Core0 data        | 1                    | 0                 |                                                      |
| Nexus_3_0         |                      | 8                 | Nexus_3_0 arbitrates with Core0 data for XBAR port 1 |
| Core1 instruction | 2                    | 1                 |                                                      |
| Core1 data        | 3                    | 1                 |                                                      |
| Nexus_3_1         |                      | 9                 | Nexus_3_1 arbitrates with Core1 data for XBAR port 3 |

Table continues on the next page...

**Table 9-1. XBAR master ports and logical master IDs (continued)**

| Module | Physical master port | Logical master ID | Comment                                                  |
|--------|----------------------|-------------------|----------------------------------------------------------|
| eDMA_A | 4                    | 4                 |                                                          |
| eDMA_B | 5                    | 5                 |                                                          |
| CSE    | 6                    | 3                 | CSE is connected to master 6 port through a concentrator |
| FEC    | 7                    | 2                 | FEC on dedicated master via 2:1 clock gasket             |

The following table lists the XBAR slave port assignments and the master ports that can access each slave port. Each of the two slave ports for the Flash Memory Controller (PFLASH) module is dedicated to a different core master port. The two slave ports for SRAM support concurrent data accesses to the SRAM.

**Table 9-2. XBAR slave port assignments**

| Module                       | Slave port | Select Address                      | Accessible by master port numbers |
|------------------------------|------------|-------------------------------------|-----------------------------------|
| PFLASH                       | 0          | Address[31:29] = 000b               | 0,1 <sup>1</sup>                  |
| PRAMC_0                      | 2          | Address[31:18] = 0100_0000_0000_00b | 0,1,2,3,4,5,6,7                   |
| PFLASH                       | 3          | Address[31:29] = 000b               | 2,3,4,5,6,7                       |
| PRAMC_1                      | 4          | Address[31:18] = 0100_0000_0000_01b | 0,1,2,3,4,5,6,7                   |
| —                            | 5          | —                                   | —                                 |
| AIPS_1 (Peripheral Bridge A) | 6          | Address[31:29] = 110b               | 0,1,2,3,4,5,6,7                   |
| AIPS_0 (Peripheral Bridge B) | 7          | Address[31:29] = 111b               | 0,1,2,3,4,5,6,7                   |

1. Master ports 2,3,4,5,6,7 access the flash memory module through slave port 3.

### 9.1.2 Unimplemented PRS<sub>n</sub> and CRS<sub>n</sub> registers

On this chip, the PRS5 and CRS5 registers are not implemented. In [Memory Map / Register Definition](#), the locations of these registers are reserved and should not be accessed.

### 9.1.3 Reset value of PRS<sub>n</sub> and CRS<sub>n</sub> registers

The reset value of the XBAR\_PRS<sub>n</sub> registers is 7654\_3210h.

The reset value of the XBAR\_CRS<sub>n</sub> registers is 00FF\_0000h.

## 9.1.4 Master priority elevation

Each implemented XBAR slave has a corresponding [Control Register \(XBAR\\_CRSn\)](#) whose number is the same as the slave's physical port. Each of these registers has HPE<sub>n</sub> fields corresponding to the physical port numbers of the implemented XBAR masters.

In general, set the HPE<sub>n</sub> field for a particular master to 1 to allow the elevation of that master's priority of access to the applicable slave.

### 9.1.4.1 Master priority elevation not used by CSE or FEC

The CSE is connected to XBAR master physical port 6, and the FEC is connected to XBAR master physical port 7. The CSE and FEC cannot signal a request for priority elevation, regardless of the value of the HPE6 and HPE7 fields in every [Control Register \(XBAR\\_CRSn\)](#).

### 9.1.4.2 Additional programming for some masters' priority elevation

Enabling the elevation of priority requires additional programming for some masters. The following table summarizes this information. The table refers to fields in the HID1 register that [Hardware Implementation Dependent Register 1 \(HID1\)](#) defines further.

**Table 9-3. Additional programming for priority elevation**

| Master            | XBAR physical port | Additional programming                                                                                                                                                                                                                                                                                         | Result                                                                                                                                                                                                                                                           |
|-------------------|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Core0 instruction | 0                  | In addition to setting HPE0 to 1 in XBAR_CRSn for the applicable slave, set to 1 either or both of these fields in the special-purpose HID1 register for core0: <ul style="list-style-type: none"> <li>Elevate External Exceptions (EEE), bit 22</li> <li>Elevate Critical Exceptions (ECE), bit 23</li> </ul> | Core0's priority is elevated when an event occurs that corresponds to the HID1 field or fields whose value is 1: <ul style="list-style-type: none"> <li>External exception when HID1[EEE] is 1</li> <li>Critical or NMI exception when HID1[ECE] is 1</li> </ul> |
| Core0 data        | 1                  | In addition to setting HPE1 to 1 in XBAR_CRSn for the applicable slave, set to 1 either or both of these fields in the special-purpose HID1 register for core0: <ul style="list-style-type: none"> <li>Elevate External Exceptions (EEE), bit 22</li> <li>Elevate Critical Exceptions (ECE), bit 23</li> </ul> | Core0's priority is elevated when an event occurs that corresponds to the HID1 field or fields whose value is 1: <ul style="list-style-type: none"> <li>External exception when HID1[EEE] is 1</li> <li>Critical or NMI exception when HID1[ECE] is 1</li> </ul> |
| Core1 instruction | 2                  | In addition to setting HPE2 to 1 in XBAR_CRSn for the applicable slave, set to 1 either or both of these fields in the special-purpose HID1 register for core1:                                                                                                                                                | Core1's priority is elevated when an event occurs that corresponds to the HID1 field or fields whose value is 1:                                                                                                                                                 |

*Table continues on the next page...*

**Table 9-3. Additional programming for priority elevation (continued)**

| Master     | XBAR physical port | Additional programming                                                                                                                                                                                                                                                                                         | Result                                                                                                                                                                                                                                                           |
|------------|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|            |                    | <ul style="list-style-type: none"> <li>Elevate External Exceptions (EEE), bit 22</li> <li>Elevate Critical Exceptions (ECE), bit 23</li> </ul>                                                                                                                                                                 | <ul style="list-style-type: none"> <li>External exception when HID1[EEE] is 1</li> <li>Critical or NMI exception when HID1[ECE] is 1</li> </ul>                                                                                                                  |
| Core1 data | 3                  | In addition to setting HPE3 to 1 in XBAR_CRSn for the applicable slave, set to 1 either or both of these fields in the special-purpose HID1 register for core1: <ul style="list-style-type: none"> <li>Elevate External Exceptions (EEE), bit 22</li> <li>Elevate Critical Exceptions (ECE), bit 23</li> </ul> | Core1's priority is elevated when an event occurs that corresponds to the HID1 field or fields whose value is 1: <ul style="list-style-type: none"> <li>External exception when HID1[EEE] is 1</li> <li>Critical or NMI exception when HID1[ECE] is 1</li> </ul> |
| eDMA_A     | 4                  | In addition to setting HPE4 to 1 in XBAR_CRSn for the applicable slave, set eDMA_A's TCD[BWC] field to 01b.                                                                                                                                                                                                    | Only eDMA_A can elevate its priority for the applicable slave.                                                                                                                                                                                                   |
| eDMA_B     | 5                  | In addition to setting HPE5 to 1 in XBAR_CRSn for the applicable slave, set eDMA_B's TCD[BWC] field to 01b.                                                                                                                                                                                                    | Only eDMA_B can elevate its priority for the applicable slave.                                                                                                                                                                                                   |

## Hardware Implementation Dependent Register 1 (HID1)

The following HID1 details supplement and replace information in the *e200z759n3 Core Reference Manual* (document number e200z759n3CRM).

HID1 is the core register (SPR 1009) that controls XBAR priority elevation.

|       |          |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |    |  |
|-------|----------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|----|----|----|--|
|       | 0        | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25       | 26 | 27 | 28 | 29 | 30 | 31 |  |
| R     | Reserved |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    | E  | E  | A  | Reserved |    |    |    |    |    |    |  |
| W     |          |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    | E  | E  | T  |          |    |    |    |    |    |    |  |
| Reset |          |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    | E  | E  | S  |          |    |    |    |    |    |    |  |
|       | 0        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0        | 0  | 0  | 0  | 0  | 0  | 0  |  |

**Table 9-4. HID1 field descriptions**

| Field | Name     | Description                                                                                                                                                          |
|-------|----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0-21  | Reserved | This field is reserved.                                                                                                                                              |
| 22    | EEE      | Elevate External Exceptions<br>0 Disable XBAR priority elevation for normal, external exceptions<br>1 Enable XBAR priority elevation for normal, external exceptions |
| 23    | ECE      | Elevate Critical Exceptions<br>0 Disable XBAR priority elevation for NMI and critical exceptions<br>1 Enable XBAR priority elevation for NMI and critical exceptions |
| 24    | ATS      | Atomic Status (read-only)<br>See the <i>e200z759n3 Core Reference Manual</i> (document number e200z759n3CRM).                                                        |
| 25-31 | Reserved | This field is reserved.                                                                                                                                              |

## 9.2 Introduction

The information found here provides information on the layout, configuration, and programming of the crossbar switch.

The crossbar switch connects bus masters and bus slaves using a crossbar switch structure. This structure allows all bus masters to access different bus slaves simultaneously, while providing arbitration among the bus masters when they access the same slave. A variety of bus arbitration methods and attributes may be programmed on a slave-by-slave basis.

### 9.2.1 Features

The crossbar switch includes these features:

- Symmetric crossbar bus switch implementation
  - Allows concurrent accesses from different masters to different slaves
  - Slave arbitration attributes configured on a slave-by-slave basis
- Up to single-clock 64-bit transfer
- Support for burst transfers of up to 16 beats of data
- Low-Power Park mode support
- Master priority elevation

## 9.3 Memory Map / Register Definition

Each slave port of the crossbar switch contains configuration registers. Read- and write-transfers require two bus clock cycles. The registers can be read from and written to only in supervisor mode. Additionally, these registers can be read from or written to only by 32-bit accesses.

A bus error response is returned if an unimplemented location is accessed within the crossbar switch.

The  $CRS_n$  and  $PRS_n$  registers can be programmed to be read-only to prevent changes to their configuration. After being read-only protected, future writes to them will terminate with an error.

**NOTE**

This section shows the registers for all eight master and slave ports. If a master or slave is not used on this particular device, then unexpected results occur when writing to its registers. See the chip configuration details for the exact master/slave assignments for your device.

All references to the crossbar switch registers are based on the physical port connections.

**XBAR memory map**

| Address offset (hex) | Register name                        | Width (in bits) | Access | Reset value                 | Section/page              |
|----------------------|--------------------------------------|-----------------|--------|-----------------------------|---------------------------|
| 0                    | Priority Registers Slave (XBAR_PRS0) | 32              | R/W    | <a href="#">See section</a> | <a href="#">9.3.1/412</a> |
| 10                   | Control Register (XBAR_CRS0)         | 32              | R/W    | <a href="#">See section</a> | <a href="#">9.3.2/415</a> |
| 100                  | Priority Registers Slave (XBAR_PRS1) | 32              | R/W    | <a href="#">See section</a> | <a href="#">9.3.1/412</a> |
| 110                  | Control Register (XBAR_CRS1)         | 32              | R/W    | <a href="#">See section</a> | <a href="#">9.3.2/415</a> |
| 200                  | Priority Registers Slave (XBAR_PRS2) | 32              | R/W    | <a href="#">See section</a> | <a href="#">9.3.1/412</a> |
| 210                  | Control Register (XBAR_CRS2)         | 32              | R/W    | <a href="#">See section</a> | <a href="#">9.3.2/415</a> |
| 300                  | Priority Registers Slave (XBAR_PRS3) | 32              | R/W    | <a href="#">See section</a> | <a href="#">9.3.1/412</a> |
| 310                  | Control Register (XBAR_CRS3)         | 32              | R/W    | <a href="#">See section</a> | <a href="#">9.3.2/415</a> |
| 400                  | Priority Registers Slave (XBAR_PRS4) | 32              | R/W    | <a href="#">See section</a> | <a href="#">9.3.1/412</a> |
| 410                  | Control Register (XBAR_CRS4)         | 32              | R/W    | <a href="#">See section</a> | <a href="#">9.3.2/415</a> |
| 500                  | Priority Registers Slave (XBAR_PRS5) | 32              | R/W    | <a href="#">See section</a> | <a href="#">9.3.1/412</a> |
| 510                  | Control Register (XBAR_CRS5)         | 32              | R/W    | <a href="#">See section</a> | <a href="#">9.3.2/415</a> |
| 600                  | Priority Registers Slave (XBAR_PRS6) | 32              | R/W    | <a href="#">See section</a> | <a href="#">9.3.1/412</a> |
| 610                  | Control Register (XBAR_CRS6)         | 32              | R/W    | <a href="#">See section</a> | <a href="#">9.3.2/415</a> |
| 700                  | Priority Registers Slave (XBAR_PRS7) | 32              | R/W    | <a href="#">See section</a> | <a href="#">9.3.1/412</a> |
| 710                  | Control Register (XBAR_CRS7)         | 32              | R/W    | <a href="#">See section</a> | <a href="#">9.3.2/415</a> |

**9.3.1 Priority Registers Slave (XBAR\_PRSn)**

The priority registers (PRSn) set the priority of each master port on a per slave port basis and reside in each slave port. The priority register can be accessed only with 32-bit accesses. After the CRSn[RO] bit is set, the PRSn register can only be read; attempts to write to it have no effect on PRSn and result in a bus-error response to the master initiating the write.

Two available masters must not be programmed with the same priority level. Attempts to program two or more masters with the same priority level result in a bus-error response and the PRSn is not updated.



**NOTE**

Valid values for the  $M_n$  priority fields depend on which masters are available on the chip. This information can be found in the chip-specific information for the crossbar.

- If the chip contains fewer than three masters, only one bit is valid.
- If the chip contains fewer than five masters, only two bits are valid.
- If five or more masters are present, all three bits of the priority field are used.

**NOTE**

See the chip-specific crossbar information for the reset value of this register.

Address:  $0h \text{ base} + 0h \text{ offset} + (256d \times i)$ , where  $i=0d \text{ to } 7d$

| Bit   | 0 | 1  | 2  | 3  | 4 | 5  | 6  | 7  | 8 | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
|-------|---|----|----|----|---|----|----|----|---|----|----|----|----|----|----|----|
| R     | 0 |    |    |    | 0 |    |    |    | 0 |    |    |    | 0  |    |    |    |
| W     |   |    |    |    |   |    |    |    |   |    |    |    |    |    |    |    |
| Reset | 0 | 0* | 0* | 0* | 0 | 0* | 0* | 0* | 0 | 0* | 0* | 0* | 0  | 0* | 0* | 0* |

| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| R     | 0  |    |    |    | 0  |    |    |    | 0  |    |    |    | 0  |    |    |    |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0  | 0* | 0* | 0* | 0  | 0* | 0* | 0* | 0  | 0* | 0* | 0* | 0  | 0* | 0* | 0* |

\* Notes:

•

**XBAR\_PRSn field descriptions**

| Field         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 1–3<br>M7     | Master 7 Priority. Sets the arbitration priority for this port on the associated slave port.<br><br>000 This master has level 1, or highest, priority when accessing the slave port.<br>001 This master has level 2 priority when accessing the slave port.<br>010 This master has level 3 priority when accessing the slave port.<br>011 This master has level 4 priority when accessing the slave port.<br>100 This master has level 5 priority when accessing the slave port.<br>101 This master has level 6 priority when accessing the slave port.<br>110 This master has level 7 priority when accessing the slave port.<br>111 This master has level 8, or lowest, priority when accessing the slave port. |
| 4<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 5–7<br>M6     | Master 6 Priority. Sets the arbitration priority for this port on the associated slave port.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |

*Table continues on the next page...*

**XBAR\_PRSn field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | 000 This master has level 1, or highest, priority when accessing the slave port.<br>001 This master has level 2 priority when accessing the slave port.<br>010 This master has level 3 priority when accessing the slave port.<br>011 This master has level 4 priority when accessing the slave port.<br>100 This master has level 5 priority when accessing the slave port.<br>101 This master has level 6 priority when accessing the slave port.<br>110 This master has level 7 priority when accessing the slave port.<br>111 This master has level 8, or lowest, priority when accessing the slave port.                                                                                                 |
| 8<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 9–11<br>M5     | Master 5 Priority. Sets the arbitration priority for this port on the associated slave port.<br>000 This master has level 1, or highest, priority when accessing the slave port.<br>001 This master has level 2 priority when accessing the slave port.<br>010 This master has level 3 priority when accessing the slave port.<br>011 This master has level 4 priority when accessing the slave port.<br>100 This master has level 5 priority when accessing the slave port.<br>101 This master has level 6 priority when accessing the slave port.<br>110 This master has level 7 priority when accessing the slave port.<br>111 This master has level 8, or lowest, priority when accessing the slave port. |
| 12<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 13–15<br>M4    | Master 4 Priority. Sets the arbitration priority for this port on the associated slave port.<br>000 This master has level 1, or highest, priority when accessing the slave port.<br>001 This master has level 2 priority when accessing the slave port.<br>010 This master has level 3 priority when accessing the slave port.<br>011 This master has level 4 priority when accessing the slave port.<br>100 This master has level 5 priority when accessing the slave port.<br>101 This master has level 6 priority when accessing the slave port.<br>110 This master has level 7 priority when accessing the slave port.<br>111 This master has level 8, or lowest, priority when accessing the slave port. |
| 16<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 17–19<br>M3    | Master 3 Priority. Sets the arbitration priority for this port on the associated slave port.<br>000 This master has level 1, or highest, priority when accessing the slave port.<br>001 This master has level 2 priority when accessing the slave port.<br>010 This master has level 3 priority when accessing the slave port.<br>011 This master has level 4 priority when accessing the slave port.<br>100 This master has level 5 priority when accessing the slave port.<br>101 This master has level 6 priority when accessing the slave port.<br>110 This master has level 7 priority when accessing the slave port.<br>111 This master has level 8, or lowest, priority when accessing the slave port. |
| 20<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |

*Table continues on the next page...*

**XBAR\_PRSn field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 21–23<br>M2    | <p>Master 2 Priority. Sets the arbitration priority for this port on the associated slave port.</p> <p>000 This master has level 1, or highest, priority when accessing the slave port.</p> <p>001 This master has level 2 priority when accessing the slave port.</p> <p>010 This master has level 3 priority when accessing the slave port.</p> <p>011 This master has level 4 priority when accessing the slave port.</p> <p>100 This master has level 5 priority when accessing the slave port.</p> <p>101 This master has level 6 priority when accessing the slave port.</p> <p>110 This master has level 7 priority when accessing the slave port.</p> <p>111 This master has level 8, or lowest, priority when accessing the slave port.</p> |
| 24<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 25–27<br>M1    | <p>Master 1 Priority. Sets the arbitration priority for this port on the associated slave port.</p> <p>000 This master has level 1, or highest, priority when accessing the slave port.</p> <p>001 This master has level 2 priority when accessing the slave port.</p> <p>010 This master has level 3 priority when accessing the slave port.</p> <p>011 This master has level 4 priority when accessing the slave port.</p> <p>100 This master has level 5 priority when accessing the slave port.</p> <p>101 This master has level 6 priority when accessing the slave port.</p> <p>110 This master has level 7 priority when accessing the slave port.</p> <p>111 This master has level 8, or lowest, priority when accessing the slave port.</p> |
| 28<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 29–31<br>M0    | <p>Master 0 Priority. Sets the arbitration priority for this port on the associated slave port.</p> <p>000 This master has level 1, or highest, priority when accessing the slave port.</p> <p>001 This master has level 2 priority when accessing the slave port.</p> <p>010 This master has level 3 priority when accessing the slave port.</p> <p>011 This master has level 4 priority when accessing the slave port.</p> <p>100 This master has level 5 priority when accessing the slave port.</p> <p>101 This master has level 6 priority when accessing the slave port.</p> <p>110 This master has level 7 priority when accessing the slave port.</p> <p>111 This master has level 8, or lowest, priority when accessing the slave port.</p> |

**9.3.2 Control Register (XBAR\_CRSn)**

These registers control several features of each slave port and must be accessed using 32-bit accesses. After CRSn[RO] is set, the PRSn can only be read; attempts to write to it have no effect and result in an error response.

**NOTE**

See the chip-specific crossbar information for the reset value of this register.

Not all HPE<sub>n</sub> fields may be active. See the chip-specific crossbar information for which masters support master high-priority elevation. Setting a field corresponding to a master that does not support master high-priority elevation has no effect.

Address: 0h base + 10h offset + (256d × i), where i=0d to 7d

|       |    |     |    |    |    |    |    |    |      |      |      |      |      |      |      |      |
|-------|----|-----|----|----|----|----|----|----|------|------|------|------|------|------|------|------|
| Bit   | 0  | 1   | 2  | 3  | 4  | 5  | 6  | 7  | 8    | 9    | 10   | 11   | 12   | 13   | 14   | 15   |
| R     | RO | HLP | 0  |    |    |    |    |    | HPE7 | HPE6 | HPE5 | HPE4 | HPE3 | HPE2 | HPE1 | HPE0 |
| W     |    |     |    |    |    |    |    |    |      |      |      |      |      |      |      |      |
| Reset | 0* | 0*  | 0* | 0* | 0* | 0* | 0* | 0* | 0*   | 0*   | 0*   | 0*   | 0*   | 0*   | 0*   | 0*   |

|       |    |    |    |    |    |    |     |    |    |    |      |    |    |      |    |    |
|-------|----|----|----|----|----|----|-----|----|----|----|------|----|----|------|----|----|
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22  | 23 | 24 | 25 | 26   | 27 | 28 | 29   | 30 | 31 |
| R     | 0  |    |    |    |    |    | ARB |    | 0  |    | PCTL |    | 0  | PARK |    |    |
| W     |    |    |    |    |    |    |     |    |    |    |      |    |    |      |    |    |
| Reset | 0* | 0* | 0* | 0* | 0* | 0* | 0*  | 0* | 0* | 0* | 0*   | 0* | 0* | 0*   | 0* | 0* |

\* Notes:

•

### XBAR\_CRSn field descriptions

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>RO         | Read Only<br><br>Forces the PRS <sub>n</sub> and CRS <sub>n</sub> registers to be read-only. After being set, only a hardware reset clears this field.<br><br>0 The CRS <sub>n</sub> and PRS <sub>n</sub> registers are writeable<br>1 The CRS <sub>n</sub> and PRS <sub>n</sub> registers are read-only and cannot be written (attempted writes have no effect on the registers and result in a bus error response).                   |
| 1<br>HLP        | Halt Low Priority<br><br>Sets the initial arbitration priority for low power mode requests . Setting this bit will not affect the request for low power mode from attaining highest priority once it has control of the slave ports.<br><br>0 The low power mode request has the highest priority for arbitration on this slave port<br>1 The low power mode request has the lowest initial priority for arbitration on this slave port |
| 2–7<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                 |
| 8<br>HPE7       | On this slave port, enable master high-priority elevation for master 7. If enabled, the master is able to elevate its priority to the highest.<br><br>0 Priority elevation for master 7 is disabled on this slave port<br>1 Priority elevation for master 7 is enabled on this slave port                                                                                                                                               |
| 9<br>HPE6       | On this slave port, enable master high-priority elevation for master 6. If enabled, the master is able to elevate its priority to the highest.<br><br>0 Priority elevation for master 6 is disabled on this slave port<br>1 Priority elevation for master 6 is enabled on this slave port                                                                                                                                               |
| 10<br>HPE5      | On this slave port, enable master high-priority elevation for master 5. If enabled, the master is able to elevate its priority to the highest.                                                                                                                                                                                                                                                                                          |

Table continues on the next page...

**XBAR\_CRSn field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | 0 Priority elevation for master 5 is disabled on this slave port<br>1 Priority elevation for master 5 is enabled on this slave port                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 11<br>HPE4        | On this slave port, enable master high-priority elevation for master 4. If enabled, the master is able to elevate its priority to the highest.<br><br>0 Priority elevation for master 4 is disabled on this slave port<br>1 Priority elevation for master 4 is enabled on this slave port                                                                                                                                                                                                                                                                                           |
| 12<br>HPE3        | On this slave port, enable master high-priority elevation for master 3. If enabled, the master is able to elevate its priority to the highest.<br><br>0 Priority elevation for master 3 is disabled on this slave port<br>1 Priority elevation for master 3 is enabled on this slave port                                                                                                                                                                                                                                                                                           |
| 13<br>HPE2        | On this slave port, enable master high-priority elevation for master 2. If enabled, the master is able to elevate its priority to the highest.<br><br>0 Priority elevation for master 2 is disabled on this slave port<br>1 Priority elevation for master 2 is enabled on this slave port                                                                                                                                                                                                                                                                                           |
| 14<br>HPE1        | On this slave port, enable master high-priority elevation for master 1. If enabled, the master is able to elevate its priority to the highest.<br><br>0 Priority elevation for master 1 is disabled on this slave port<br>1 Priority elevation for master 1 is enabled on this slave port                                                                                                                                                                                                                                                                                           |
| 15<br>HPE0        | On this slave port, enable master high-priority elevation for master 0. If enabled, the master is able to elevate its priority to the highest.<br><br>0 Priority elevation for master 0 is disabled on this slave port<br>1 Priority elevation for master 0 is enabled on this slave port                                                                                                                                                                                                                                                                                           |
| 16–21<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 22–23<br>ARB      | Arbitration Mode<br><br>Selects the arbitration policy for the slave port.<br><br>00 Fixed priority<br>01 Round-robin (rotating) priority<br>10 Reserved<br>11 Reserved                                                                                                                                                                                                                                                                                                                                                                                                             |
| 24–25<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 26–27<br>PCTL     | Parking Control<br><br>Determines the slave port's parking control. The low-power park feature results in an overall power savings if the slave port is not saturated; however, this forces an extra latency clock when any master tries to access the slave port while not in use because it is not parked on any master.<br><br>00 When no master makes a request, the arbiter parks the slave port on the master port defined by the PARK bit field<br>01 When no master makes a request, the arbiter parks the slave port on the last master to be in control of the slave port |

*Table continues on the next page...*

**XBAR\_CRSn field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | 10 Low-power park. When no master makes a request, the slave port is not parked on a master and the arbiter drives all outputs to a constant safe state<br>11 Reserved                                                                                                                                                                                                                                                                                                                                             |
| 28<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 29–31<br>PARK  | Park<br>Determines which master port the current slave port parks on when no masters are actively making requests and the PCTL bits are cleared.<br><b>NOTE:</b> Select only master ports that are present on the chip. Otherwise, undefined behavior might occur.<br>000 Park on master port M0<br>001 Park on master port M1<br>010 Park on master port M2<br>011 Park on master port M3<br>100 Park on master port M4<br>101 Park on master port M5<br>110 Park on master port M6<br>111 Park on master port M7 |

## 9.4 Functional Description

### 9.4.1 General operation

When a master accesses the crossbar switch, the access is immediately taken. If the targeted slave port of the access is available, then the access is immediately presented on the slave port. Single-clock or zero-wait-state accesses are possible through the crossbar. If the targeted slave port of the access is busy or parked on a different master port, the requesting master simply sees wait states inserted until the targeted slave port can service the master's request. The latency in servicing the request depends on each master's priority level and the responding slave's access time.

Because the crossbar switch appears to be just another slave to the master device, the master device has no knowledge of whether it actually owns the slave port it is targeting. While the master does not have control of the slave port it is targeting, it simply waits.

After the master has control of the slave port it is targeting, the master remains in control of the slave port until it relinquishes the slave port by running an IDLE cycle or by targeting a different slave port for its next access.

The master can also lose control of the slave port if another higher-priority master makes a request to the slave port. However, if the master is running a fixed-length burst transfer it retains control of the slave port until that transfer completes.

The crossbar terminates all master IDLE transfers, as opposed to allowing the termination to come from one of the slave buses. Additionally, when no master is requesting access to a slave port, the crossbar drives IDLE transfers onto the slave bus, even though a default master may be granted access to the slave port.

When a slave bus is being idled by the crossbar, it can park the slave port on the master port indicated by  $CRS_n[PARK]$ . This is done to save the initial clock of arbitration delay that otherwise would be seen if the same master had to arbitrate to gain control of the slave port. The slave port can also be put into Low Power Park mode to save power, by using  $CRS_n[PCTL]$ .

## 9.4.2 Register coherency

The operation of the crossbar is affected as soon as a register is written. The values of the registers do not track with slave-port-related master accesses, but instead track only with slave accesses.

## 9.4.3 Arbitration

The crossbar switch supports two arbitration algorithms:

- Fixed priority
- Round-robin

The arbitration scheme is independently programmable for each slave port.

### 9.4.3.1 Fixed-priority operation

When operating in fixed-priority mode, each master is assigned a unique priority level in the priority registers ( $PRS_n$ ). If two masters request access to the same slave port, the master with the highest priority in the selected priority register gains control over the slave port.

#### NOTE

In this arbitration mode, a higher-priority master can monopolize a slave port, preventing accesses from any lower-priority master to the port.

When a master makes a request to a slave port, the slave port checks whether the new requesting master's priority level is higher than that of the master that currently has control over the slave port, unless the slave port is in a parked state. The slave port performs an arbitration check at every clock edge to ensure that the proper master, if any, has control of the slave port.

The following table describes possible scenarios based on the requesting master port:

**Table 9-24. How the Crossbar Switch grants control of a slave port to a master**

| When                                                                                                                                                                                                                                                           | Then the Crossbar Switch grants control to the requesting master                                                                                                                            |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Both of the following are true: <ul style="list-style-type: none"> <li>The current master is not running a transfer.</li> <li>The new requesting master's priority level is higher than that of the current master.</li> </ul>                                 | At the next clock edge                                                                                                                                                                      |
| Both of the following are true: <ul style="list-style-type: none"> <li>The current master is running a fixed length burst transfer or a locked transfer.</li> <li>The requesting master's priority level is higher than that of the current master.</li> </ul> | At the end of the burst transfer or locked transfer                                                                                                                                         |
| The requesting master's priority level is lower than the current master.                                                                                                                                                                                       | At the conclusion of one of the following cycles: <ul style="list-style-type: none"> <li>An IDLE cycle</li> <li>A non-IDLE cycle to a location other than the current slave port</li> </ul> |

### 9.4.3.2 Round-robin priority operation

When operating in round-robin mode, each master is assigned a relative priority based on the master port number. This relative priority is compared to the master port number (ID) of the last master to perform a transfer on the slave bus. The highest priority requesting master becomes owner of the slave bus at the next transfer boundary, accounting for locked and fixed-length burst transfers. Priority is based on how far ahead the ID of the requesting master is to the ID of the last master.

After granted access to a slave port, a master may perform as many transfers as desired to that port until another master makes a request to the same slave port. The next master in line is granted access to the slave port at the next transfer boundary, or possibly on the next clock cycle if the current master has no pending access request.

As an example of arbitration in round-robin mode, assume the crossbar is implemented with master ports 0, 1, 4, and 5. If the last master of the slave port was master 1, and master 0, 4, and 5 make simultaneous requests, they are serviced in the order: 4 then 5 then 0.



Parking may continue to be used in a round-robin mode, but does not affect the round-robin pointer unless the parked master actually performs a transfer. Handoff occurs to the next master in line after one cycle of arbitration. If the slave port is put into low-power park mode, the round-robin pointer is reset to point at master port 0, giving it the highest priority.

### 9.4.3.3 Priority assignment

Each master port must be assigned a unique 3-bit priority level. If an attempt is made to program multiple master ports with the same priority level within the priority registers (PRSn), the crossbar switch responds with a bus error and the registers are not updated.

## 9.5 Initialization/application information

No initialization is required for the crossbar switch.

Hardware reset ensures that all the register bits used by the crossbar switch are properly initialized to a valid state. However, settings and priorities may be programmed to achieve maximum system performance.

### NOTE

- During configuration of the crossbar switch, all other masters must be idle.
- To prevent reconfiguration of the crossbar switch, write 1 to CRSn[RO].



# Chapter 10

## Crossbar Integrity Checker (XBIC)

### 10.1 Chip-specific XBIC information

This chip has one instance of the XBIC module.

#### 10.1.1 XBIC master and slave assignments

The XBIC identifies each XBAR master and slave in terms of the master or slave's physical port number. See the "Physical master port" assignments in [Table 9-1](#) and the "Slave port" assignments in [Table 9-2](#).

#### 10.1.2 Unimplemented MCR and ESR fields

On this chip, the MCR[SE5] and ESR[DPSE5] fields are not implemented. In [XBIC Module Control Register \(XBIC\\_MCR\)](#) and [XBIC Error Status Register \(XBIC\\_ESR\)](#), these fields are reserved.

### 10.2 Overview

The Crossbar Integrity Checker (XBIC) verifies the integrity of the crossbar transfers.

### 10.3 Features

The XBIC has the following features:

- Verification of attribute information for all crossbar transfers

- EDC (72,64) code protects against single and double bit errors
- Verification of feedback information for each data phase during crossbar transfer
- Error injection for testing
  - Programmable master and slave port specifiers
  - Programmable 8-bit toggle vector to insert error in master EDC checkbit value
  - Address, EDC syndrome, master and slave port information captured on error
- Programmable integrity check enable on a per-slave-port basis
- Programmable integrity feedback check enable on a per-master-port basis

## 10.4 Block diagram

The crossbar transfer attribute information for all master and slave ports is routed to the XBIC, which calculates and checks the EDC parity over the attribute information as shown in the following diagram.

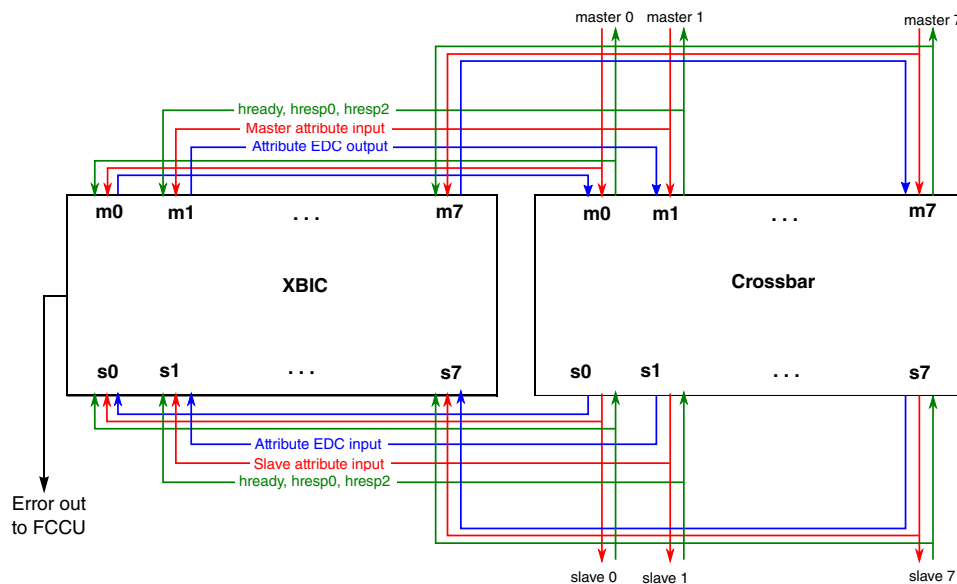


Figure 10-1. XBIC system block diagram

## 10.5 External signal description

The XBIC has no external interface signals.

## 10.6 Memory map and register definition

The XBIC programming model has four 32-bit registers. The programming model can only be accessed in supervisor mode using 32-bit (word) accesses. Attempted references with a different access size, to an undefined (reserved) address, with a non-supported access type (a write to a read-only register), or in user mode generate an error termination.

**XBIC memory map**

| Address offset (hex) | Register name                            | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|------------------------------------------|-----------------|--------|-------------|----------------------------|
| 0                    | XBIC Module Control Register (XBIC_MCR)  | 32              | R/W    | FFFF_0000h  | <a href="#">10.6.1/425</a> |
| 4                    | XBIC Error Injection Register (XBIC_EIR) | 32              | R/W    | 0000_0000h  | <a href="#">10.6.2/427</a> |
| 8                    | XBIC Error Status Register (XBIC_ESR)    | 32              | R      | 0000_0000h  | <a href="#">10.6.3/428</a> |
| C                    | XBIC Error Address Register (XBIC_EAR)   | 32              | R      | 0000_0000h  | <a href="#">10.6.4/430</a> |

### 10.6.1 XBIC Module Control Register (XBIC\_MCR)

The XBIC\_MCR allows attribute integrity checking to be enabled on a per-port basis.

Address: 0h base + 0h offset = 0h

|       |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit   | 0   | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | 10  | 11  | 12  | 13  | 14  | 15  |
| R     | SE0 | SE1 | SE2 | SE3 | SE4 | SE5 | SE6 | SE7 | ME0 | ME1 | ME2 | ME3 | ME4 | ME5 | ME6 | ME7 |
| W     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| Reset | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   |
| Bit   | 16  | 17  | 18  | 19  | 20  | 21  | 22  | 23  | 24  | 25  | 26  | 27  | 28  | 29  | 30  | 31  |
| R     | 0   |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| W     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
| Reset | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |

**XBIC\_MCR field descriptions**

| Field    | Description                                                                                                                                                      |
|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>SE0 | Slave Port Enable for EDC error detect.<br>0 Attribute integrity checking disabled for slave port 0.<br>1 Attribute integrity checking enabled for slave port 0. |
| 1<br>SE1 | Slave Port Enable for EDC error detect.<br>0 Attribute integrity checking disabled for slave port 1.<br>1 Attribute integrity checking enabled for slave port 1. |

*Table continues on the next page...*

**XBIC\_MCR field descriptions (continued)**

| Field     | Description                                                                                                                                                                         |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2<br>SE2  | Slave Port Enable for EDC error detect.<br>0 Attribute integrity checking disabled for slave port 2.<br>1 Attribute integrity checking enabled for slave port 2.                    |
| 3<br>SE3  | Slave Port Enable for EDC error detect.<br>0 Attribute integrity checking disabled for slave port 3.<br>1 Attribute integrity checking enabled for slave port 3.                    |
| 4<br>SE4  | Slave Port Enable for EDC error detect.<br>0 Attribute integrity checking disabled for slave port 4.<br>1 Attribute integrity checking enabled for slave port 4.                    |
| 5<br>SE5  | Slave Port Enable for EDC error detect.<br>0 Attribute integrity checking disabled for slave port 5.<br>1 Attribute integrity checking enabled for slave port 5.                    |
| 6<br>SE6  | Slave Port Enable for EDC error detect.<br>0 Attribute integrity checking disabled for slave port 6.<br>1 Attribute integrity checking enabled for slave port 6.                    |
| 7<br>SE7  | Slave Port Enable for EDC error detect.<br>0 Attribute integrity checking disabled for slave port 7.<br>1 Attribute integrity checking enabled for slave port 7.                    |
| 8<br>ME0  | Master Port Enable for slave driven signal safety check.<br>0 Attribute integrity checking disabled for master port 0.<br>1 Attribute integrity checking enabled for master port 0. |
| 9<br>ME1  | Master Port Enable for slave driven signal safety check.<br>0 Attribute integrity checking disabled for master port 1.<br>1 Attribute integrity checking enabled for master port 1. |
| 10<br>ME2 | Master Port Enable for slave driven signal safety check.<br>0 Attribute integrity checking disabled for master port 2.<br>1 Attribute integrity checking enabled for master port 2. |
| 11<br>ME3 | Master Port Enable for slave driven signal safety check.<br>0 Attribute integrity checking disabled for master port 3.<br>1 Attribute integrity checking enabled for master port 3. |
| 12<br>ME4 | Master Port Enable for slave driven signal safety check.<br>0 Attribute integrity checking disabled for master port 4.<br>1 Attribute integrity checking enabled for master port 4. |
| 13<br>ME5 | Master Port Enable for slave driven signal safety check.<br>0 Attribute integrity checking disabled for master port 5.<br>1 Attribute integrity checking enabled for master port 5. |

*Table continues on the next page...*

**XBIC\_MCR field descriptions (continued)**

| Field             | Description                                                                                                                                                                         |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 14<br>ME6         | Master Port Enable for slave driven signal safety check.<br>0 Attribute integrity checking disabled for master port 6.<br>1 Attribute integrity checking enabled for master port 6. |
| 15<br>ME7         | Master Port Enable for slave driven signal safety check.<br>0 Attribute integrity checking disabled for master port 7.<br>1 Attribute integrity checking enabled for master port 7. |
| 16–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                             |

**10.6.2 XBIC Error Injection Register (XBIC\_EIR)**

The XBIC\_EIR contains fields for controlling the XBIC error injection function. When an error is injected, the EDC syndrome associated with the programmed master and slave ports is modified, causing the XBIC to assert the error indication to the FCCU and capturing the transfer information in the XBIC\_ESR and XBIC\_EAR registers.

Otherwise, transfers are not affected by XBIC error injection.

Address: 0h base + 4h offset = 4h

|       |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0   | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | EIE |    |    |    |    |    |    |    | 0  |    |    |    |    |    |    |    |
| W     |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| Bit   | 16  | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**XBIC\_EIR field descriptions**

| Field            | Description                                                                                                                                                                              |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>EIE         | Error Injection Enable.<br>0 Error injection disabled<br>1 Error injection enabled                                                                                                       |
| 1–16<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                  |
| 17–19<br>SLV     | Target Slave Port. Error injection is enabled for the designated slave port. Other slave ports are unaffected.                                                                           |
| 20–23<br>MST     | Target Master ID. Error injection is enabled for the designated master ID. Transfers with other master IDs are not affected.                                                             |
| 24–31<br>SYN     | Syndrome. This value is exclusive-ored with the calculated syndrome (which should be zero) to generate an error with the specified syndrome. A value of zero does not generate an error. |

10.6.3 XBIC Error Status Register (XBIC\_ESR)

The Error Status Register (XBIC\_ESR) contains two types of error information about the last transfer. If an attribute integrity check error was detected, the slave port, the master port, and the syndrome are captured in the SLV, MST, and SYN fields. If there is a mismatch among internal signals (hready, hresp0, and hresp2) during the data phase, the slave and master port are catured in the DPSE0-7 and DPME0-7 fields.

This register is cleared only on reset.

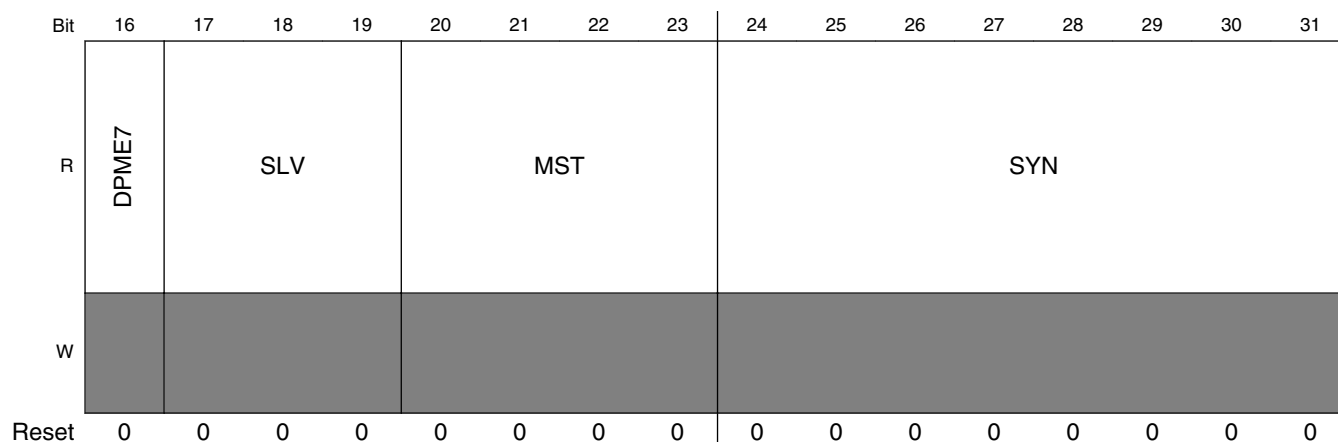
NOTE

When the XBIC detects back-to-back faults on a system bus path through the crossbar switch, the fault information captured in the XBIC\_ESR does not correspond to the initial fault event but rather the subsequent fault event. While the fault event is properly detected, diagnostic status information in the XBIC\_ESR register describing the initial fault event is lost. This can only occur in the event of a series of bus transactions targeting the same crossbar slave target where the series of bus transactions are not separated by idle or stall cycles.

Address: 0h base + 8h offset = 8h

| Bit   | 0   | 1     | 2     | 3     | 4     | 5     | 6     | 7     | 8     | 9     | 10    | 11    | 12    | 13    | 14    | 15    |
|-------|-----|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| R     | VLD | DPSE0 | DPSE1 | DPSE2 | DPSE3 | DPSE4 | DPSE5 | DPSE6 | DPSE7 | DPME0 | DPME1 | DPME2 | DPME3 | DPME4 | DPME5 | DPME6 |
| W     |     |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| Reset | 0   | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |



**XBIC\_ESR field descriptions**

| Field      | Description                                                                                                                                              |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>VLD   | Error Status Valid.<br>0 No error detected, other fields of the ESR and EAR are not valid.<br>1 Error detected, all fields of the ESR and EAR are valid. |
| 1<br>DPSE0 | Data phase slave port error.<br>0 No error on slave port 0<br>1 Data phase error on slave port 0                                                         |
| 2<br>DPSE1 | Data phase slave port error.<br>0 No error on slave port 1<br>1 Data phase error on slave port 1                                                         |
| 3<br>DPSE2 | Data phase slave port error.<br>0 No error on slave port 2<br>1 Data phase error on slave port 2                                                         |
| 4<br>DPSE3 | Data phase slave port error.<br>0 No error on slave port 3<br>1 Data phase error on slave port 3                                                         |
| 5<br>DPSE4 | Data phase slave port error.<br>0 No error on slave port 4<br>1 Data phase error on slave port 4                                                         |
| 6<br>DPSE5 | Data phase slave port error.<br>0 No error on slave port 5<br>1 Data phase error on slave port 5                                                         |
| 7<br>DPSE6 | Data phase slave port error.<br>0 No error on slave port 6<br>1 Data phase error on slave port 6                                                         |
| 8<br>DPSE7 | Data phase slave port error.                                                                                                                             |

*Table continues on the next page...*

**XBIC\_ESR field descriptions (continued)**

| Field        | Description                                                                                                                                                         |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|              | 0 No error on slave port 7<br>1 Data phase error on slave port 7                                                                                                    |
| 9<br>DPME0   | Data phase master port error.<br>0 No error on master port 0<br>1 Data phase error on master port 0                                                                 |
| 10<br>DPME1  | Data phase master port error.<br>0 No error on master port 1<br>1 Data phase error on master port 1                                                                 |
| 11<br>DPME2  | Data phase master port error.<br>0 No error on master port 2<br>1 Data phase error on master port 2                                                                 |
| 12<br>DPME3  | Data phase master port error.<br>0 No error on master port 3<br>1 Data phase error on master port 3                                                                 |
| 13<br>DPME4  | Data phase master port error.<br>0 No error on master port 4<br>1 Data phase error on master port 4                                                                 |
| 14<br>DPME5  | Data phase master port error.<br>0 No error on master port 5<br>1 Data phase error on master port 5                                                                 |
| 15<br>DPME6  | Data phase master port error.<br>0 No error on master port 6<br>1 Data phase error on master port 6                                                                 |
| 16<br>DPME7  | Data phase master port error.<br>0 No error on master port 7<br>1 Data phase error on master port 7                                                                 |
| 17–19<br>SLV | Slave Port. The slave port targeted by the last transfer with an error.                                                                                             |
| 20–23<br>MST | Master ID. The master ID value of the last transfer with an error.                                                                                                  |
| 24–31<br>SYN | Syndrom. The syndrome calculated for the last transfer with an error. For single bit errors the signal in error can be determined, see <a href="#">Table 10-6</a> . |

**10.6.4 XBIC Error Address Register (XBIC\_EAR)**

The Error Address Register (XBIC\_EAR) contains the address of the first transfer in which an attribute integrity check error was detected on an enabled slave port. This register is only cleared on reset.

**NOTE**

When the XBIC detects back-to-back faults on a system bus path through the crossbar switch, the fault information captured in the XBIC\_EAR does not correspond to the initial fault event but rather the subsequent fault event. While the fault event is properly detected, diagnostic status information in the XBIC\_ESR register describing the initial fault event is lost. This can only occur in the event of a series of bus transactions targeting the same crossbar slave target where the series of bus transactions are not separated by idle or stall cycles.

Address: 0h base + Ch offset = Ch

|       |      |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | ADDR |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |      |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0    | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**XBIC\_EAR field descriptions**

| Field        | Description                                     |
|--------------|-------------------------------------------------|
| 0–31<br>ADDR | The address of the last transfer with an error. |

## 10.7 Functional description

The Crossbar Integrity Checker (XBIC) verifies the integrity of the crossbar interface. The Module Control Register (XBIC\_MCR) allows integrity checking to be enabled on a individual port basis. When an error is detected, it is captured in XBIC\_ESR register and XBIC also sends out an error signal to the Fault Collection and Control Unit (FCCU). Detection of an error does not generate a bus error. The XBIC integrity checking is independent from the end-to-end ECC that covers the transfer address and data.

During the AHB address phase, the crossbar attribute information is checked using an 8-bit Error Detection Code (EDC), which detects any single- or double-bit errors. When an error is detected, due either to hardware fault or error injection, information related to the error is captured and reported in XBIC\_ESR and XBIC\_EAR. The cause of a single-bit error can be determined by the syndrome value reported in the XBIC\_ESR as shown in [Table 10-6](#). Any other syndrome indicates a multi-bit error.

During the AHB data phase, the responding signals from the slave to master are checked when passing through the crossbar. When an error is detected, the information is captured in the XBIC\_ESR DPSE0-7 and DPME0-7 fields.

During the AHB data phase, the XBIC will additionally send an alarm to the FCCU if a master port attempts a back-to-back access where the first access terminates normally, but the second access attempts to access an address space that is not mapped to any slave on the Crossbar (AXBS). The resultant 'absent slave error' causes the AXBS to generate a bus error response to the requesting master that the XBIC will detect as a difference since the bus error did not originate from the slave port.

The XBIC can also be programmed to intentionally inject EDC errors for testing. Error injection is targeted toward a single slave port and a single master ID at a time as determined by the settings in the Error Injection Register (XBIC\_EIR). When an error is injected, the EDC syndrome is modified which causes the XBIC to assert the error indication to the FCCU. Otherwise transfers are unaffected by XBIC error injection. This approach allows the check logic to be verified without compromising the integrity of the data transfer. Once the error injection function is enabled by setting XBIC\_EIR[EIE], errors are induced on all subsequent targeted transactions until XBIC\_EIR[EIE] is cleared

**Table 10-6. Hexadecimal attribute single-bit error syndromes**

| Signal    | SYN | Signal     | SYN | Signal     | SYN | Signal     | SYN |
|-----------|-----|------------|-----|------------|-----|------------|-----|
| hwrite    | 07  | hbstrb[7]  | 70  | hdecor[31] | 25  | hdecor[15] | 23  |
| htrans[0] | 0b  | hbstrb[6]  | 32  | hdecor[30] | 68  | hdecor[14] | 51  |
| hsize[2]  | 0d  | hbstrb[5]  | 52  | hdecor[29] | c7  | hdecor[13] | 54  |
| hsize[1]  | 0e  | hbstrb[4]  | a8  | hdecor[28] | 83  | hdecor[12] | 61  |
| hsize[0]  | 13  | hbstrb[3]  | 43  | hdecor[27] | 85  | hdecor[11] | e3  |
| hprot[5]  | 15  | hbstrb[2]  | 45  | hdecor[26] | 86  | hdecor[10] | e6  |
| hprot[4]  | 16  | hbstrb[1]  | 4c  | hdecor[25] | 89  | hdecor[9]  | f8  |
| hprot[3]  | 19  | hbstrb[0]  | a4  | hdecor[24] | 8a  | hdecor[8]  | 38  |
| hprot[2]  | 1a  | hmaster[3] | a2  | hdecor[23] | 8c  | hdecor[7]  | 58  |
| hprot[1]  | 1c  | hmaster[2] | b0  | hdecor[22] | 49  | hdecor[6]  | 37  |
| hprot[0]  | 91  | hmaster[1] | c1  | hdecor[21] | 92  | hdecor[5]  | f1  |
| hburst[2] | a1  | hmaster[0] | c2  | hdecor[20] | 94  | hdecor[4]  | 3b  |
| hburst[1] | 64  | hslave[2]  | c4  | hdecor[19] | 98  | hdecor[3]  | 3d  |
| hburst[0] | 29  | hslave[1]  | c8  | hdecor[18] | 46  | hdecor[2]  | 3e  |
| hmastlock | 2a  | hslave[0]  | d0  | hdecor[17] | 34  | hdecor[1]  | 4f  |
| hunalign  | 2c  | hdecorated | e0  | hdecor[16] | 4a  | hdecor[0]  | 6e  |
| edc[7]    | 80  | edc[6]     | 40  | edc[5]     | 20  | edc[4]     | 10  |
| edc[3]    | 08  | edc[2]     | 04  | edc[1]     | 02  | edc[0]     | 01  |

# Chapter 11

## Peripheral Bridge (AIPS/PBRIDGE)

### 11.1 Chip-specific AIPS/PBRIDGE information

This chip has two instances of the AIPS (PBRIDGE) module:

- AIPS\_0 (PBRIDGE\_B)
- AIPS\_1 (PBRIDGE\_A)

#### 11.1.1 Modules assigned to PACR $n$ and OPACR $n$ fields

Each module attached to the Peripheral Bridges is assigned to a memory map slot corresponding to an AIPS register field.

- Every on-platform module has an assigned PACR $n$  field within the PACRA to PACRD registers.
- Every off-platform module has an assigned OPACR $n$  field within the OPACRA to OPACRH registers.

The [Peripheral Access Control Register \(AIPS\\_PACR \$n\$ \)](#) description provides the location of each PACR $n$  field, and the [Off-Platform Peripheral Access Control Register \(AIPS\\_OPACR \$n\$ \)](#) description provides the location of each OPACR $n$  field.

The following tables provide the module slot assignments for this chip.

#### NOTE

Unused PACR $n$  and OPACR $n$  fields are reserved.

**Table 11-1. AIPS\_0 (PBRIDGE\_B) PACR<sub>n</sub> field assignments**

| Module  | PACR field |
|---------|------------|
| AIPS_0  | 0          |
| XBAR    | 1          |
| XBIC    | 2          |
| MPU_0   | 3          |
| MPU_1   | 4          |
| CSE     | 5          |
| TDM     | 6          |
| PRAMC_0 | 7          |
| PRAMC_1 | 8          |
| SEMA4   | 9          |
| SWT_B   | 13         |
| SWT_A   | 14         |
| STM_A   | 15         |
| eDMA_A  | 17         |
| INTC    | 18         |
| FEC     | 19         |
| PCU     | 20         |
| eDMA_B  | 21         |
| ERM     | 24         |
| EIM     | 25         |
| PFLASH  | 26         |
| PCM     | 27         |

**Table 11-2. AIPS\_0 (PBRIDGE\_B) OPACR<sub>n</sub> field assignments**

| Module    | OPACR |
|-----------|-------|
| EQADC_A   | 0     |
| DSPI_A    | 4     |
| DSPI_B    | 5     |
| DSPI_C    | 6     |
| ESCI_A    | 12    |
| ESCI_B    | 13    |
| ESCI_C    | 14    |
| FlexCAN_A | 16    |
| FlexCAN_B | 17    |
| C55FMC    | 26    |
| PASS      | 29    |

*Table continues on the next page...*

**Table 11-2. AIPS\_0 (PBRIDGE\_B) OPACR<sub>n</sub> field assignments  
(continued)**

| Module          | OPACR |
|-----------------|-------|
| SSCM            | 30    |
| BAM             | 31    |
| ETPU_C          | 32    |
| ETPU_C PARAM    | 34    |
| ETPU_C PARAM    | 35    |
| ETPU_C CODE     | 36    |
| GLITCH_FILTER_0 | 40    |
| M_CAN_0         | 44    |
| M_CAN RAM       | 45    |
| M_CAN_1         | 46    |
| SDADC_2         | 49    |
| SDADC_4         | 50    |
| SRX_0           | 55    |
| EMIOS_0         | 57    |
| CRC             | 58    |
| PCS             | 60    |
| PLLDIG          | 61    |
| DTS             | 62    |
| JDC             | 63    |

**Table 11-3. AIPS\_1 (PBRIDGE\_A) PACR<sub>n</sub> field assignments**

| Module | PACR |
|--------|------|
| AIPS_1 | 0    |

**Table 11-4. AIPS\_1 (PBRIDGE\_A) OPACR<sub>n</sub> field assignments**

| Peripheral | OPACR |
|------------|-------|
| FCCU       | 0     |
| SIU_A      | 4     |
| SIU_B      | 6     |
| EMIOS_1    | 8     |
| ESCI_D     | 12    |
| ESCI_E     | 13    |
| ESCI_F     | 14    |
| PMU        | 15    |

*Table continues on the next page...*

**Table 11-4. AIPS\_1 (PBRIDGE\_A) OPACR<sub>n</sub> field assignments  
(continued)**

| Peripheral      | OPACR |
|-----------------|-------|
| ETPU_A/B        | 16    |
| ETPU_A/B PARAM  | 18    |
| ETPU_A/B PARAM  | 19    |
| ETPU_A/B CODE   | 20    |
| ETPU_A/B CODE   | 21    |
| DSPI_D          | 23    |
| DSPI_E          | 24    |
| PIT_RTI         | 28    |
| STCU2           | 29    |
| GLITCH_FILTER_1 | 33    |
| SDADC_1         | 49    |
| SDADC_3         | 50    |
| EQADC_B         | 53    |
| SRX_1           | 55    |
| FlexCAN_C       | 56    |
| FlexCAN_D       | 57    |
| CMU_0 to CMU_8  | 59    |

### 11.1.2 AIPS/PBRIDGE register reset values

The following table identifies chip-specific reset values of AIPS/PBRIDGE registers.

**Table 11-5. AIPS/PBRIDGE register reset values**

| Register                           | AIPS_0 (PBRIDGE_B) | AIPS_1 (PBRIDGE_A) |
|------------------------------------|--------------------|--------------------|
| MPRA                               | 7777_7777h         | 7777_7777h         |
| MPRB                               | 7777_7777h         | 7777_7777h         |
| PACRA                              | 5444_4444h         | 5000_0000h         |
| PACRB                              | 4400_0444h         | 0000_0000h         |
| PACRC                              | 0444_4400h         | 0000_0000h         |
| PACRD                              | 4444_0000h         | 0000_0000h         |
| OPACRA, OPACRB, and OPACRC         | 4444_4444h         | 4444_4444h         |
| OPACRD                             | 4454_4444h         | 4444_4444h         |
| OPACRE, OPACRF, OPACRG, and OPACRH | 4444_4444h         | 4444_4444h         |



## 11.2 Introduction

The peripheral bridge converts the crossbar switch interface to an interface that can access most of the slave peripherals on this chip.

The peripheral bridge occupies 64 MB of the address space, which is divided into peripheral slots of 16 KB. (It might be possible that all the peripheral slots are not used. See the memory map chapter for details on slot assignments.) The bridge includes separate clock enable inputs for each of the slots to accommodate slower peripherals.

### 11.2.1 Features

Key features of the peripheral bridge are:

- Supports peripheral slots with 8-, 16-, and 32-bit datapath width
- Supports a pair of 32-bit transactions for selected 64-bit memory accesses
- Programming model provides memory protection functionality

### 11.2.2 General operation

The slave devices connected to the peripheral bridge are modules which contain a programming model of control and status registers. The system masters read and write these registers through the peripheral bridge.

The register maps of the peripherals are located on 16 KB boundaries. Each peripheral is allocated one or more 16-KB block(s) of the memory map.

## 11.3 Memory map/register definition

The 32-bit peripheral bridge registers can be accessed only in supervisor mode by trusted bus masters. Additionally, these registers must be read from or written to only by a 32-bit aligned access. The peripheral bridge registers are mapped into the Peripheral Access Control Register A PACRA[PACR0] address space.

## AIPS memory map

| Address offset (hex) | Register name                                                 | Width (in bits) | Access | Reset value                 | Section/ page              |
|----------------------|---------------------------------------------------------------|-----------------|--------|-----------------------------|----------------------------|
| 0                    | Master Privilege Register A (AIPS_MPRA)                       | 32              | R/W    | <a href="#">See section</a> | <a href="#">11.3.1/438</a> |
| 4                    | Master Privilege Register B (AIPS_MPRB)                       | 32              | R/W    | <a href="#">See section</a> | <a href="#">11.3.2/442</a> |
| 20                   | Peripheral Access Control Register (AIPS_PACRA)               | 32              | R/W    | <a href="#">See section</a> | <a href="#">11.3.3/444</a> |
| 24                   | Peripheral Access Control Register (AIPS_PACRB)               | 32              | R/W    | <a href="#">See section</a> | <a href="#">11.3.3/444</a> |
| 28                   | Peripheral Access Control Register (AIPS_PACRC)               | 32              | R/W    | <a href="#">See section</a> | <a href="#">11.3.3/444</a> |
| 2C                   | Peripheral Access Control Register (AIPS_PACRD)               | 32              | R/W    | <a href="#">See section</a> | <a href="#">11.3.3/444</a> |
| 40                   | Off-Platform Peripheral Access Control Register (AIPS_OPACRA) | 32              | R/W    | <a href="#">See section</a> | <a href="#">11.3.4/449</a> |
| 44                   | Off-Platform Peripheral Access Control Register (AIPS_OPACRB) | 32              | R/W    | <a href="#">See section</a> | <a href="#">11.3.4/449</a> |
| 48                   | Off-Platform Peripheral Access Control Register (AIPS_OPACRC) | 32              | R/W    | <a href="#">See section</a> | <a href="#">11.3.4/449</a> |
| 4C                   | Off-Platform Peripheral Access Control Register (AIPS_OPACRD) | 32              | R/W    | <a href="#">See section</a> | <a href="#">11.3.4/449</a> |
| 50                   | Off-Platform Peripheral Access Control Register (AIPS_OPACRE) | 32              | R/W    | <a href="#">See section</a> | <a href="#">11.3.4/449</a> |
| 54                   | Off-Platform Peripheral Access Control Register (AIPS_OPACRF) | 32              | R/W    | <a href="#">See section</a> | <a href="#">11.3.4/449</a> |
| 58                   | Off-Platform Peripheral Access Control Register (AIPS_OPACRG) | 32              | R/W    | <a href="#">See section</a> | <a href="#">11.3.4/449</a> |
| 5C                   | Off-Platform Peripheral Access Control Register (AIPS_OPACRH) | 32              | R/W    | <a href="#">See section</a> | <a href="#">11.3.4/449</a> |

### 11.3.1 Master Privilege Register A (AIPS\_MPRA)

The MPRA specifies identical 4-bit fields defining the access-privilege level associated with a bus master to various peripherals on the chip. The register provides one field per bus master.

#### NOTE

At reset, the default value loaded into the MPRA fields is chip-specific. See the chip configuration details for the value of a particular device.

A register field that maps to an unimplemented master or peripheral behaves as read-only-zero.

Each master is assigned a logical ID from 0 to 15. See the master logical ID assignment table in the chip-specific AIPS information.

Address: 0h base + 0h offset = 0h

|       |    |      |      |      |    |      |      |      |    |      |      |      |    |      |      |      |
|-------|----|------|------|------|----|------|------|------|----|------|------|------|----|------|------|------|
| Bit   | 0  | 1    | 2    | 3    | 4  | 5    | 6    | 7    | 8  | 9    | 10   | 11   | 12 | 13   | 14   | 15   |
| R     | 0  | MTR0 | MTW0 | MPL0 | 0  | MTR1 | MTW1 | MPL1 | 0  | MTR2 | MTW2 | MPL2 | 0  | MTR3 | MTW3 | MPL3 |
| W     |    |      |      |      |    |      |      |      |    |      |      |      |    |      |      |      |
| Reset | 0* | 0*   | 0*   | 0*   | 0* | 0*   | 0*   | 0*   | 0* | 0*   | 0*   | 0*   | 0* | 0*   | 0*   | 0*   |

|       |    |      |      |      |    |      |      |      |    |      |      |      |    |          |    |    |
|-------|----|------|------|------|----|------|------|------|----|------|------|------|----|----------|----|----|
| Bit   | 16 | 17   | 18   | 19   | 20 | 21   | 22   | 23   | 24 | 25   | 26   | 27   | 28 | 29       | 30 | 31 |
| R     | 0  | MTR4 | MTW4 | MPL4 | 0  | MTR5 | MTW5 | MPL5 | 0  | MTR6 | MTW6 | MPL6 | 0  | Reserved |    |    |
| W     |    |      |      |      |    |      |      |      |    |      |      |      |    |          |    |    |
| Reset | 0* | 0*   | 0*   | 0*   | 0* | 0*   | 0*   | 0*   | 0* | 0*   | 0*   | 0*   | 0* | 0*       | 0* | 0* |

\* Notes:

•

**AIPS\_MPRA field descriptions**

| Field         | Description                                                                                                                                                                                                            |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                |
| 1<br>MTR0     | Master 0 Trusted For Read<br><br>Determines whether the master is trusted for read accesses.<br><br>0 This master is not trusted for read accesses.<br>1 This master is trusted for read accesses.                     |
| 2<br>MTW0     | Master 0 Trusted For Writes<br><br>Determines whether the master is trusted for write accesses.<br><br>0 This master is not trusted for write accesses.<br>1 This master is trusted for write accesses.                |
| 3<br>MPL0     | Master 0 Privilege Level<br><br>Specifies how the privilege level of the master is determined.<br><br>0 Accesses from this master are forced to user-mode.<br>1 Accesses from this master are not forced to user-mode. |
| 4<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                |
| 5<br>MTR1     | Master 1 Trusted for Read<br><br>Determines whether the master is trusted for read accesses.<br><br>0 This master is not trusted for read accesses.<br>1 This master is trusted for read accesses.                     |
| 6<br>MTW1     | Master 1 Trusted for Writes<br><br>Determines whether the master is trusted for write accesses.                                                                                                                        |

*Table continues on the next page...*

**AIPS\_MPRA field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                        |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | 0 This master is not trusted for write accesses.<br>1 This master is trusted for write accesses.                                                                                                                   |
| 7<br>MPL1      | Master 1 Privilege Level<br>Specifies how the privilege level of the master is determined.<br><br>0 Accesses from this master are forced to user-mode.<br>1 Accesses from this master are not forced to user-mode. |
| 8<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                            |
| 9<br>MTR2      | Master 2 Trusted For Read<br>Determines whether the master is trusted for read accesses.<br><br>0 This master is not trusted for read accesses.<br>1 This master is trusted for read accesses.                     |
| 10<br>MTW2     | Master 2 Trusted For Writes<br>Determines whether the master is trusted for write accesses.<br><br>0 This master is not trusted for write accesses.<br>1 This master is trusted for write accesses.                |
| 11<br>MPL2     | Master 2 Privilege Level<br>Specifies how the privilege level of the master is determined.<br><br>0 Accesses from this master are forced to user-mode.<br>1 Accesses from this master are not forced to user-mode. |
| 12<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                            |
| 13<br>MTR3     | Master 3 Trusted For Read<br>Determines whether the master is trusted for read accesses.<br><br>0 This master is not trusted for read accesses.<br>1 This master is trusted for read accesses.                     |
| 14<br>MTW3     | Master 3 Trusted For Writes<br>Determines whether the master is trusted for write accesses.<br><br>0 This master is not trusted for write accesses.<br>1 This master is trusted for write accesses.                |
| 15<br>MPL3     | Master 3 Privilege Level<br>Specifies how the privilege level of the master is determined.<br><br>0 Accesses from this master are forced to user-mode.<br>1 Accesses from this master are not forced to user-mode. |
| 16<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                            |

*Table continues on the next page...*

**AIPS\_MPRA field descriptions (continued)**

| <b>Field</b>   | <b>Description</b>                                                                                                                                                                                                     |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 17<br>MTR4     | Master 4 Trusted For Read<br><br>Determines whether the master is trusted for read accesses.<br><br>0 This master is not trusted for read accesses.<br>1 This master is trusted for read accesses.                     |
| 18<br>MTW4     | Master 4 Trusted For Writes<br><br>Determines whether the master is trusted for write accesses.<br><br>0 This master is not trusted for write accesses.<br>1 This master is trusted for write accesses.                |
| 19<br>MPL4     | Master 4 Privilege Level<br><br>Specifies how the privilege level of the master is determined.<br><br>0 Accesses from this master are forced to user-mode.<br>1 Accesses from this master are not forced to user-mode. |
| 20<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                |
| 21<br>MTR5     | Master 5 Trusted For Read<br><br>Determines whether the master is trusted for read accesses.<br><br>0 This master is not trusted for read accesses.<br>1 This master is trusted for read accesses.                     |
| 22<br>MTW5     | Master 5 Trusted For Writes<br><br>Determines whether the master is trusted for write accesses.<br><br>0 This master is not trusted for write accesses.<br>1 This master is trusted for write accesses.                |
| 23<br>MPL5     | Master 5 Privilege Level<br><br>Specifies how the privilege level of the master is determined.<br><br>0 Accesses from this master are forced to user-mode.<br>1 Accesses from this master are not forced to user-mode. |
| 24<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                |
| 25<br>MTR6     | Master 6 Trusted for Read<br><br>Determines whether the master is trusted for read accesses.<br><br>0 This master is not trusted for read accesses.<br>1 This master is trusted for read accesses.                     |
| 26<br>MTW6     | Master 6 Trusted for Writes<br><br>Determines whether the master is trusted for write accesses.<br><br>0 This master is not trusted for write accesses.<br>1 This master is trusted for write accesses.                |

*Table continues on the next page...*

**AIPS\_MPRB field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                        |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 27<br>MPL6        | Master 6 Privilege Level<br>Specifies how the privilege level of the master is determined.<br><br>0 Accesses from this master are forced to user-mode.<br>1 Accesses from this master are not forced to user-mode. |
| 28<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                            |
| 29–31<br>Reserved | This field is reserved.                                                                                                                                                                                            |

**11.3.2 Master Privilege Register B (AIPS\_MPRB)**

The MPRB registers specifies identical 4-bit fields defining the access-privilege level associated with a bus master to the various peripherals on the chip. The register provides one field per bus master.

**NOTE**

At reset, the default value loaded into the MPRB fields is chip-specific. See the chip configuration information for the value on your particular device.

Accesses to registers or register fields which correspond to master or peripheral locations which are not implemented return zeros on reads, and are ignored on writes.

Each master is assigned a logical ID from 0 to 15. See the master logical ID assignment table in the AIPS chip-specific information.

Address: 0h base + 4h offset = 4h

| Bit   | 0  | 1    | 2    | 3    | 4  | 5    | 6    | 7    | 8  | 9        | 10 | 11 | 12 | 13       | 14 | 15 |
|-------|----|------|------|------|----|------|------|------|----|----------|----|----|----|----------|----|----|
| R     | 0  |      |      |      | 0  |      |      |      | 0  |          |    |    | 0  |          |    |    |
| W     |    | MTR8 | MTW8 | MPL8 |    | MTR9 | MTW9 | MPL9 |    | Reserved |    |    |    | Reserved |    |    |
| Reset | 0* | 0*   | 0*   | 0*   | 0* | 0*   | 0*   | 0*   | 0* | 0*       | 0* | 0* | 0* | 0*       | 0* | 0* |

| Bit   | 16 | 17       | 18 | 19 | 20 | 21       | 22 | 23 | 24 | 25       | 26 | 27 | 28 | 29       | 30 | 31 |
|-------|----|----------|----|----|----|----------|----|----|----|----------|----|----|----|----------|----|----|
| R     | 0  |          |    |    | 0  |          |    |    | 0  |          |    |    | 0  |          |    |    |
| W     |    | Reserved |    |    |    | Reserved |    |    |    | Reserved |    |    |    | Reserved |    |    |
| Reset | 0* | 0*       | 0* | 0* | 0* | 0*       | 0* | 0* | 0* | 0*       | 0* | 0* | 0* | 0*       | 0* | 0* |

\* Notes:

•

**AIPS\_MPRB field descriptions**

| Field             | Description                                                                                                                                                                                                            |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>Reserved     | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                |
| 1<br>MTR8         | Master 8 Trusted for Read<br><br>Determines whether the master is trusted for read accesses.<br><br>0 This master is not trusted for read accesses.<br>1 This master is trusted for read accesses.                     |
| 2<br>MTW8         | Master 8 Trusted for Writes<br><br>Determines whether the master is trusted for write accesses.<br><br>0 This master is not trusted for write accesses.<br>1 This master is trusted for write accesses.                |
| 3<br>MPL8         | Master 8 Privilege Level<br><br>Specifies how the privilege level of the master is determined.<br><br>0 Accesses from this master are forced to user-mode.<br>1 Accesses from this master are not forced to user-mode. |
| 4<br>Reserved     | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                |
| 5<br>MTR9         | Master 9 Trusted for Read<br><br>Determines whether the master is trusted for read accesses.<br><br>0 This master is not trusted for read accesses.<br>1 This master is trusted for read accesses.                     |
| 6<br>MTW9         | Master 9 Trusted for Writes<br><br>Determines whether the master is trusted for write accesses.<br><br>0 This master is not trusted for write accesses.<br>1 This master is trusted for write accesses.                |
| 7<br>MPL9         | Master 9 Privilege Level<br><br>Specifies how the privilege level of the master is determined.<br><br>0 Accesses from this master are forced to user-mode.<br>1 Accesses from this master are not forced to user-mode. |
| 8<br>Reserved     | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                |
| 9–11<br>Reserved  | This field is reserved.                                                                                                                                                                                                |
| 12<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                |
| 13–15<br>Reserved | This field is reserved.                                                                                                                                                                                                |
| 16<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                |

*Table continues on the next page...*

**AIPS\_MPRB field descriptions (continued)**

| Field             | Description                                                                             |
|-------------------|-----------------------------------------------------------------------------------------|
| 17–19<br>Reserved | This field is reserved.                                                                 |
| 20<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 21–23<br>Reserved | This field is reserved.                                                                 |
| 24<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 25–27<br>Reserved | This field is reserved.                                                                 |
| 28<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 29–31<br>Reserved | This field is reserved.                                                                 |

**11.3.3 Peripheral Access Control Register (AIPS\_PACRn)**

Each PACR register consists of eight 4-bit PACR fields. Each PACR field defines the access levels for a particular on-platform peripheral. The peripheral assignment to each PACR field is defined by the memory map slot of the peripheral. See the chip-specific AIPS information for the field assignment of a particular peripheral.

Every PACR field to which no peripheral is assigned is reserved. Reads to reserved locations return zeros, and writes are ignored.

The following table shows the location of each peripheral slot's PACR field in the PACR registers.

| Offset | Register | [0:3]  | [4:7]  | [8:11] | [12:15] | [16:19] | [20:23] | [24:27] | [28:31] |
|--------|----------|--------|--------|--------|---------|---------|---------|---------|---------|
| 0x20   | PACRA    | PACR0  | PACR1  | PACR2  | PACR3   | PACR4   | PACR5   | PACR6   | PACR7   |
| 0x24   | PACRB    | PACR8  | PACR9  | PACR10 | PACR11  | PACR12  | PACR13  | PACR14  | PACR15  |
| 0x28   | PACRC    | PACR16 | PACR17 | PACR18 | PACR19  | PACR20  | PACR21  | PACR22  | PACR23  |
| 0x2C   | PACRD    | PACR24 | PACR25 | PACR26 | PACR27  | PACR28  | PACR29  | PACR30  | PACR31  |

Address: 0h base + 20h offset + (4d × i), where i=0d to 3d

| Bit   | 0  | 1   | 2   | 3   | 4  | 5   | 6   | 7   | 8  | 9   | 10  | 11  | 12 | 13  | 14  | 15  |
|-------|----|-----|-----|-----|----|-----|-----|-----|----|-----|-----|-----|----|-----|-----|-----|
| R     | 0  | SP0 | WP0 | TP0 | 0  | SP1 | WP1 | TP1 | 0  | SP2 | WP2 | TP2 | 0  | SP3 | WP3 | TP3 |
| W     |    |     |     |     |    |     |     |     |    |     |     |     |    |     |     |     |
| Reset | 0* | 0*  | 0*  | 0*  | 0* | 0*  | 0*  | 0*  | 0* | 0*  | 0*  | 0*  | 0* | 0*  | 0*  | 0*  |



|       |    |     |     |     |    |     |     |     |    |     |     |     |    |     |     |     |
|-------|----|-----|-----|-----|----|-----|-----|-----|----|-----|-----|-----|----|-----|-----|-----|
| Bit   | 16 | 17  | 18  | 19  | 20 | 21  | 22  | 23  | 24 | 25  | 26  | 27  | 28 | 29  | 30  | 31  |
| R     | 0  | SP4 | WP4 | TP4 | 0  | SP5 | WP5 | TP5 | 0  | SP6 | WP6 | TP6 | 0  | SP7 | WP7 | TP7 |
| W     |    |     |     |     |    |     |     |     |    |     |     |     |    |     |     |     |
| Reset | 0* | 0*  | 0*  | 0*  | 0* | 0*  | 0*  | 0*  | 0* | 0*  | 0*  | 0*  | 0* | 0*  | 0*  | 0*  |

\* Notes:

•

**AIPS\_PACR<sub>n</sub> field descriptions**

| Field         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 1<br>SP0      | Supervisor Protect<br><br>Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPRx[MPL <sub>n</sub> ] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates.<br><br>0 This peripheral does not require supervisor privilege level for accesses.<br>1 This peripheral requires supervisor privilege level for accesses. |
| 2<br>WP0      | Write Protect<br><br>Determines whether the peripheral allows write access. When this bit is set and a write access is attempted, access terminates with an error response and no peripheral access initiates.<br><br>0 This peripheral allows write accesses.<br>1 This peripheral is write protected.                                                                                                                                                                                                                                  |
| 3<br>TP0      | Trusted Protect<br><br>Determines whether the peripheral allows accesses from an untrusted master. When this field is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates.<br><br>0 Accesses from an untrusted master are allowed.<br>1 Accesses from an untrusted master are not allowed.                                                                                                                                                            |
| 4<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 5<br>SP1      | Supervisor Protect<br><br>Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPRx[MPL <sub>n</sub> ] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates.<br><br>0 This peripheral does not require supervisor privilege level for accesses.<br>1 This peripheral requires supervisor privilege level for accesses. |
| 6<br>WP1      | Write Protect<br><br>Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates.                                                                                                                                                                                                                                                                                                                       |

*Table continues on the next page...*

**AIPS\_PACR<sub>n</sub> field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | 0 This peripheral allows write accesses.<br>1 This peripheral is write protected.                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 7<br>TP1       | Trusted Protect<br><br>Determines whether the peripheral allows accesses from an untrusted master. When this bit is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates.<br><br>0 Accesses from an untrusted master are allowed.<br>1 Accesses from an untrusted master are not allowed.                                                                                                                                                                           |
| 8<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 9<br>SP2       | Supervisor Protect<br><br>Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPR <sub>x</sub> [MPL <sub>n</sub> ] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates.<br><br>0 This peripheral does not require supervisor privilege level for accesses.<br>1 This peripheral requires supervisor privilege level for accesses. |
| 10<br>WP2      | Write Protect<br><br>Determines whether the peripheral allows write access. When this bit is set and a write access is attempted, access terminates with an error response and no peripheral access initiates.<br><br>0 This peripheral allows write accesses.<br>1 This peripheral is write protected.                                                                                                                                                                                                                                               |
| 11<br>TP2      | Trusted Protect<br><br>Determines whether the peripheral allows accesses from an untrusted master. When this field is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates.<br><br>0 Accesses from an untrusted master are allowed.<br>1 Accesses from an untrusted master are not allowed.                                                                                                                                                                         |
| 12<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 13<br>SP3      | Supervisor Protect<br><br>Determines whether the peripheral requires supervisor privilege level for access. When this bit is set, the master privilege level must indicate the supervisor access attribute, and the MPR <sub>x</sub> [MPL <sub>n</sub> ] control bit for the master must be set. If not, access terminates with an error response and no peripheral access initiates.<br><br>0 This peripheral does not require supervisor privilege level for accesses.<br>1 This peripheral requires supervisor privilege level for accesses.       |
| 14<br>WP3      | Write Protect<br><br>Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates.                                                                                                                                                                                                                                                                                                                                    |

*Table continues on the next page...*

**AIPS\_PACR<sub>n</sub> field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | 0 This peripheral allows write accesses.<br>1 This peripheral is write protected.                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 15<br>TP3      | Trusted Protect<br><br>Determines whether the peripheral allows accesses from an untrusted master. When this bit is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates.<br><br>0 Accesses from an untrusted master are allowed.<br>1 Accesses from an untrusted master are not allowed.                                                                                                                                                                           |
| 16<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 17<br>SP4      | Supervisor Protect<br><br>Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPR <sub>x</sub> [MPL <sub>n</sub> ] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates.<br><br>0 This peripheral does not require supervisor privilege level for accesses.<br>1 This peripheral requires supervisor privilege level for accesses. |
| 18<br>WP4      | Write Protect<br><br>Determines whether the peripheral allows write access. When this bit is set and a write access is attempted, access terminates with an error response and no peripheral access initiates.<br><br>0 This peripheral allows write accesses.<br>1 This peripheral is write protected.                                                                                                                                                                                                                                               |
| 19<br>TP4      | Trusted Protect<br><br>Determines whether the peripheral allows accesses from an untrusted master. When this field is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates.<br><br>0 Accesses from an untrusted master are allowed.<br>1 Accesses from an untrusted master are not allowed.                                                                                                                                                                         |
| 20<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 21<br>SP5      | Supervisor Protect<br><br>Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPR <sub>x</sub> [MPL <sub>n</sub> ] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates.<br><br>0 This peripheral does not require supervisor privilege level for accesses.<br>1 This peripheral requires supervisor privilege level for accesses. |
| 22<br>WP5      | Write Protect<br><br>Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates.                                                                                                                                                                                                                                                                                                                                    |

*Table continues on the next page...*

**AIPS\_PACR<sub>n</sub> field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | 0 This peripheral allows write accesses.<br>1 This peripheral is write protected.                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 23<br>TP5      | Trusted Protect<br><br>Determines whether the peripheral allows accesses from an untrusted master. When this field is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates.<br><br>0 Accesses from an untrusted master are allowed.<br>1 Accesses from an untrusted master are not allowed.                                                                                                                                                                         |
| 24<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 25<br>SP6      | Supervisor Protect<br><br>Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPR <sub>x</sub> [MPL <sub>n</sub> ] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates.<br><br>0 This peripheral does not require supervisor privilege level for accesses.<br>1 This peripheral requires supervisor privilege level for accesses. |
| 26<br>WP6      | Write Protect<br><br>Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates.<br><br>0 This peripheral allows write accesses.<br>1 This peripheral is write protected.                                                                                                                                                                                                                                           |
| 27<br>TP6      | Trusted Protect<br><br>Determines whether the peripheral allows accesses from an untrusted master. When this field is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates.<br><br>0 Accesses from an untrusted master are allowed.<br>1 Accesses from an untrusted master are not allowed.                                                                                                                                                                         |
| 28<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 29<br>SP7      | Supervisor Protect<br><br>Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPR <sub>x</sub> [MPL <sub>n</sub> ] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates.<br><br>0 This peripheral does not require supervisor privilege level for accesses.<br>1 This peripheral requires supervisor privilege level for accesses. |
| 30<br>WP7      | Write Protect<br><br>Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates.                                                                                                                                                                                                                                                                                                                                    |

*Table continues on the next page...*

**AIPS\_PACR<sub>n</sub> field descriptions (continued)**

| Field     | Description                                                                                                                                                                                                                                                                                                                                                                   |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|           | 0 This peripheral allows write accesses.<br>1 This peripheral is write protected.                                                                                                                                                                                                                                                                                             |
| 31<br>TP7 | Trusted Protect<br><br>Determines whether the peripheral allows accesses from an untrusted master. When this field is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates.<br><br>0 Accesses from an untrusted master are allowed.<br>1 Accesses from an untrusted master are not allowed. |

**11.3.4 Off-Platform Peripheral Access Control Register (AIPS\_OPACR<sub>n</sub>)**

Each OPACR register consists of eight 4-bit OPACR fields. Each OPACR field defines the access levels for a particular off-platform peripheral. The peripheral assignment to each OPACR field is defined by the memory map slot of the peripheral. See the chip-specific AIPS information for the field assignment of a particular peripheral.

Every OPACR field to which no peripheral is assigned is reserved. Reads to reserved locations return zeros, and writes are ignored.

The following table shows the location of each peripheral slot's OPACR field in the OPACR registers.

| Offset | Register | [0:3]   | [4:7]   | [8:11]  | [12:15] | [16:19] | [20:23] | [24:27] | [28:31] |
|--------|----------|---------|---------|---------|---------|---------|---------|---------|---------|
| 0x40   | OPACRA   | OPACR0  | OPACR1  | OPACR2  | OPACR3  | OPACR4  | OPACR5  | OPACR6  | OPACR7  |
| 0x44   | OPACRB   | OPACR8  | OPACR9  | OPACR10 | OPACR11 | OPACR12 | OPACR13 | OPACR14 | OPACR15 |
| 0x48   | OPACRC   | OPACR16 | OPACR17 | OPACR18 | OPACR19 | OPACR20 | OPACR21 | OPACR22 | OPACR23 |
| 0x4C   | OPACRD   | OPACR24 | OPACR25 | OPACR26 | OPACR27 | OPACR28 | OPACR29 | OPACR30 | OPACR31 |
| 0x50   | OPACRE   | OPACR32 | OPACR33 | OPACR34 | OPACR35 | OPACR36 | OPACR37 | OPACR38 | OPACR39 |
| 0x54   | OPACRF   | OPACR40 | OPACR41 | OPACR42 | OPACR43 | OPACR44 | OPACR45 | OPACR46 | OPACR47 |
| 0x58   | OPACRG   | OPACR48 | OPACR49 | OPACR50 | OPACR51 | OPACR52 | OPACR53 | OPACR54 | OPACR55 |
| 0x5C   | OPACRH   | OPACR56 | OPACR57 | OPACR58 | OPACR59 | OPACR60 | OPACR61 | OPACR62 | OPACR63 |

Address: 0h base + 40h offset + (4d × i), where i=0d to 7d

| Bit   | 0  | 1   | 2   | 3   | 4  | 5   | 6   | 7   | 8  | 9   | 10  | 11  | 12 | 13  | 14  | 15  |
|-------|----|-----|-----|-----|----|-----|-----|-----|----|-----|-----|-----|----|-----|-----|-----|
| R     | 0  | SP0 | WP0 | TP0 | 0  | SP1 | WP1 | TP1 | 0  | SP2 | WP2 | TP2 | 0  | SP3 | WP3 | TP3 |
| W     |    |     |     |     |    |     |     |     |    |     |     |     |    |     |     |     |
| Reset | 0* | 0*  | 0*  | 0*  | 0* | 0*  | 0*  | 0*  | 0* | 0*  | 0*  | 0*  | 0* | 0*  | 0*  | 0*  |

## Memory map/register definition

| Bit   | 16 | 17  | 18  | 19  | 20 | 21  | 22  | 23  | 24 | 25  | 26  | 27  | 28 | 29  | 30  | 31  |
|-------|----|-----|-----|-----|----|-----|-----|-----|----|-----|-----|-----|----|-----|-----|-----|
| R     | 0  | SP4 | WP4 | TP4 | 0  | SP5 | WP5 | TP5 | 0  | SP6 | WP6 | TP6 | 0  | SP7 | WP7 | TP7 |
| W     |    |     |     |     |    |     |     |     |    |     |     |     |    |     |     |     |
| Reset | 0* | 0*  | 0*  | 0*  | 0* | 0*  | 0*  | 0*  | 0* | 0*  | 0*  | 0*  | 0* | 0*  | 0*  | 0*  |

\* Notes:

•

## AIPS\_OPACR<sub>n</sub> field descriptions

| Field         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 1<br>SP0      | Supervisor Protect<br><br>Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPRx[MPL <sub>n</sub> ] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates.<br><br>0 This peripheral does not require supervisor privilege level for accesses.<br>1 This peripheral requires supervisor privilege level for accesses. |
| 2<br>WP0      | Write Protect<br><br>Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates.<br><br>0 This peripheral allows write accesses.<br>1 This peripheral is write protected.                                                                                                                                                                                                                              |
| 3<br>TP0      | Trusted Protect<br><br>Determines whether the peripheral allows accesses from an untrusted master. When this bit is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates.<br><br>0 Accesses from an untrusted master are allowed.<br>1 Accesses from an untrusted master are not allowed.                                                                                                                                                              |
| 4<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 5<br>SP1      | Supervisor Protect<br><br>Determines whether the peripheral requires supervisor privilege level for access. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPRx[MPL <sub>n</sub> ] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates.<br><br>0 This peripheral does not require supervisor privilege level for accesses.<br>1 This peripheral requires supervisor privilege level for accesses.   |
| 6<br>WP1      | Write Protect<br><br>Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates.                                                                                                                                                                                                                                                                                                                       |

Table continues on the next page...

**AIPS\_OPACR<sub>n</sub> field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | 0 This peripheral allows write accesses.<br>1 This peripheral is write protected.                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 7<br>TP1       | Trusted Protect<br><br>Determines whether the peripheral allows accesses from an untrusted master. When this field is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates.<br><br>0 Accesses from an untrusted master are allowed.<br>1 Accesses from an untrusted master are not allowed.                                                                                                                                                            |
| 8<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 9<br>SP2       | Supervisor Protect<br><br>Determines whether the peripheral requires supervisor privilege level for access. When this bit is set, the master privilege level must indicate the supervisor access attribute, and the MPRx[MPL <sub>n</sub> ] control bit for the master must be set. If not, access terminates with an error response and no peripheral access initiates.<br><br>0 This peripheral does not require supervisor privilege level for accesses.<br>1 This peripheral requires supervisor privilege level for accesses.       |
| 10<br>WP2      | Write Protect<br><br>Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates.<br><br>0 This peripheral allows write accesses.<br>1 This peripheral is write protected.                                                                                                                                                                                                                              |
| 11<br>TP2      | Trusted Protect<br><br>Determines whether the peripheral allows accesses from an untrusted master. When this bit is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates.<br><br>0 Accesses from an untrusted master are allowed.<br>1 Accesses from an untrusted master are not allowed.                                                                                                                                                              |
| 12<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 13<br>SP3      | Supervisor Protect<br><br>Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPRx[MPL <sub>n</sub> ] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates.<br><br>0 This peripheral does not require supervisor privilege level for accesses.<br>1 This peripheral requires supervisor privilege level for accesses. |
| 14<br>WP3      | Write Protect<br><br>Determines whether the peripheral allows write access. When this bit is set and a write access is attempted, access terminates with an error response and no peripheral access initiates.                                                                                                                                                                                                                                                                                                                           |

*Table continues on the next page...*

**AIPS\_OPACR<sub>n</sub> field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | 0 This peripheral allows write accesses.<br>1 This peripheral is write protected.                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 15<br>TP3      | Trusted Protect<br><br>Determines whether the peripheral allows accesses from an untrusted master. When this field is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates.<br><br>0 Accesses from an untrusted master are allowed.<br>1 Accesses from an untrusted master are not allowed.                                                                                                                                                            |
| 16<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 17<br>SP4      | Supervisor Protect<br><br>Determines whether the peripheral requires supervisor privilege level for access. When this bit is set, the master privilege level must indicate the supervisor access attribute, and the MPRx[MPL <sub>n</sub> ] control bit for the master must be set. If not, access terminates with an error response and no peripheral access initiates.<br><br>0 This peripheral does not require supervisor privilege level for accesses.<br>1 This peripheral requires supervisor privilege level for accesses.       |
| 18<br>WP4      | Write Protect<br><br>Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates.<br><br>0 This peripheral allows write accesses.<br>1 This peripheral is write protected.                                                                                                                                                                                                                              |
| 19<br>TP4      | Trusted Protect<br><br>Determines whether the peripheral allows accesses from an untrusted master. When this bit is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates.<br><br>0 Accesses from an untrusted master are allowed.<br>1 Accesses from an untrusted master are not allowed.                                                                                                                                                              |
| 20<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 21<br>SP5      | Supervisor Protect<br><br>Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPRx[MPL <sub>n</sub> ] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates.<br><br>0 This peripheral does not require supervisor privilege level for accesses.<br>1 This peripheral requires supervisor privilege level for accesses. |
| 22<br>WP5      | Write Protect<br><br>Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates.                                                                                                                                                                                                                                                                                                                       |

*Table continues on the next page...*



**AIPS\_OPACR<sub>n</sub> field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | 0 This peripheral allows write accesses.<br>1 This peripheral is write protected.                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 23<br>TP5      | Trusted Protect<br><br>Determines whether the peripheral allows accesses from an untrusted master. When this field is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates.<br><br>0 Accesses from an untrusted master are allowed.<br>1 Accesses from an untrusted master are not allowed.                                                                                                                                                            |
| 24<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 25<br>SP6      | Supervisor Protect<br><br>Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPRx[MPL <sub>n</sub> ] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates.<br><br>0 This peripheral does not require supervisor privilege level for accesses.<br>1 This peripheral requires supervisor privilege level for accesses. |
| 26<br>WP6      | Write Protect<br><br>Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates.<br><br>0 This peripheral allows write accesses.<br>1 This peripheral is write protected.                                                                                                                                                                                                                              |
| 27<br>TP6      | Trusted Protect<br><br>Determines whether the peripheral allows accesses from an untrusted master. When this field is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates.<br><br>0 Accesses from an untrusted master are allowed.<br>1 Accesses from an untrusted master are not allowed.                                                                                                                                                            |
| 28<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 29<br>SP7      | Supervisor Protect<br><br>Determines whether the peripheral requires supervisor privilege level for accesses. When this field is set, the master privilege level must indicate the supervisor access attribute, and the MPRx[MPL <sub>n</sub> ] control field for the master must be set. If not, access terminates with an error response and no peripheral access initiates.<br><br>0 This peripheral does not require supervisor privilege level for accesses.<br>1 This peripheral requires supervisor privilege level for accesses. |
| 30<br>WP7      | Write Protect<br><br>Determines whether the peripheral allows write accesses. When this field is set and a write access is attempted, access terminates with an error response and no peripheral access initiates.                                                                                                                                                                                                                                                                                                                       |

*Table continues on the next page...*

**AIPS\_OPACR<sub>n</sub> field descriptions (continued)**

| Field     | Description                                                                                                                                                                                                                                                                                                                                                                   |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|           | 0 This peripheral allows write accesses.<br>1 This peripheral is write protected.                                                                                                                                                                                                                                                                                             |
| 31<br>TP7 | Trusted Protect<br><br>Determines whether the peripheral allows accesses from an untrusted master. When this field is set and an access is attempted by an untrusted master, the access terminates with an error response and no peripheral access initiates.<br><br>0 Accesses from an untrusted master are allowed.<br>1 Accesses from an untrusted master are not allowed. |

## 11.4 Functional description

The peripheral bridge functions as a bus protocol translator between the crossbar switch and the slave peripheral bus.

Support is provided for generating a pair of 32-bit slave accesses when performing certain 64-bit peripheral accesses.

The peripheral bridge manages all transactions destined for the attached slave devices and generates select signals for modules on the peripheral bus by decoding accesses within the attached address space.

### 11.4.1 Access support

All combinations of access size and peripheral data port width are supported. An access that is larger than the target peripheral's data width will be decomposed to multiple, smaller accesses. Bus decomposition is terminated by a transfer error caused by an access to an empty register area.

## Chapter 12

# Platform Configuration Module (PCM)

### 12.1 PCM memory map and register descriptions

The Platform Configuration Module contains three miscellaneous configuration registers for the chip. Currently, the configuration registers are related to the operation of the FEC and intelligent bus bridging gasket. The module is mapped to AIPS\_0 (PBRIDGE\_B) on-platform slot 27 with a base address of FFF6\_C000h.

#### NOTE

These registers can be accessed only in supervisor mode.

#### PCM memory map

| Address offset (hex) | Register name                                               | Width (in bits) | Access | Reset value | Section/page               |
|----------------------|-------------------------------------------------------------|-----------------|--------|-------------|----------------------------|
| 0                    | FEC Burst Optimization Master Control Register (PCM_FBOMCR) | 32              | R/W    | 0000_0000h  | <a href="#">12.1.1/456</a> |
| 4                    | Bus Bridge Configuration Register 1 (PCM_IAHB_BE1)          | 32              | R/W    | 0303_0303h  | <a href="#">12.1.2/457</a> |
| 8                    | Bus Bridge Configuration Register 2 (PCM_IAHB_BE2)          | 32              | R/W    | 0303_0303h  | <a href="#">12.1.3/460</a> |

## 12.1.1 FEC Burst Optimization Master Control Register (PCM\_FBOMCR)

This register controls FEC burst optimization behavior on the system bus.

Address: 0h base + 0h offset = 0h

|       |    |    |    |    |    |        |      |      |            |    |    |    |    |    |    |    |
|-------|----|----|----|----|----|--------|------|------|------------|----|----|----|----|----|----|----|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5      | 6    | 7    | 8          | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0  |    |    |    |    |        |      |      |            |    |    |    |    |    |    |    |
| W     |    |    |    |    |    |        |      |      |            |    |    |    |    |    |    |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0      | 0    | 0    | 0          | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21     | 22   | 23   | 24         | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0  |    |    |    |    | ACCERR | WBEN | RBEN | FXSBE[7:0] |    |    |    |    |    |    |    |
| W     |    |    |    |    |    |        |      |      |            |    |    |    |    |    |    |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0      | 0    | 0    | 0          | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

### PCM\_FBOMCR field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–20<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 21<br>ACCERR     | Accumulate Error<br><br>This field determines whether an error response for the first half of the write burst is accumulated to the second half of the write burst or discarded. To complete the burst, the FEC interface to the system bus responds by indicating that the first half of the burst completed without error before it actually writes the data so that it can fetch the second half of the write data from the FIFO. When actually written onto the system bus, the first half of the write burst can have an error. Because this half initially responded without an error to the FIFO, the error is discarded or accumulated with the error response for the second half of the burst.<br><br>0 Any error to the first half of the write burst is discarded.<br>1 Any actual error response to the first half of the write burst is accumulated in the second half's response. In other words, an error response to the first half is seen in the response to the second half, even if the second half does not contain an error. |
| 22<br>WBEN       | Global write burst enable to XBAR slave port designated by FXSBE <sub>n</sub><br><br>0 Write bursting to all XBAR slave ports is disabled.<br>1 Write bursting is enabled to any XBAR slave port whose FXSBE <sub>n</sub> bit is 1.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 23<br>RBEN       | Global read burst enable from XBAR slave port designated by FXSBE <sub>n</sub><br><br>0 Read bursting from all XBAR slave ports is disabled.<br>1 Read bursting is enabled from any XBAR slave port whose FXSBE <sub>n</sub> bit is 1.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

Table continues on the next page...

**PCM\_FBOMCR field descriptions (continued)**

| Field               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 24–31<br>FXSBE[7:0] | <p>FEC XBAR slave burst enable</p> <p>This field enables bursting by the FEC interface to the XBAR slave port controlled by each FXSBE<sub>n</sub> bit.</p> <ul style="list-style-type: none"> <li>When a particular FXSBE<sub>n</sub> bit is 1, the XBAR slave port enabled by that bit can support the bursts allowed by RBEN and WBEN. RBEN enables read bursts from the XBAR slave port, and WBEN enables write bursts to the XBAR slave port.</li> <li>When a particular FXSBE<sub>n</sub> bit is 0, the FEC interface does not burst to the XBAR slave port controlled by that FXSBE<sub>n</sub> bit.</li> </ul> <p>FXSBE<sub>n</sub> assignments to XBAR slave ports are as follows:</p> <p>FXSBE0 = Flash memory</p> <p>FXSBE1 = reserved</p> <p>FXSBE2 = RAM</p> <p>FXSBE3 = reserved</p> <p>FXSBE4 = reserved</p> <p>FXSBE5 = reserved</p> <p>FXSBE6 = AIPS_1 (PBRIDGE_A)</p> <p>FXSBE7 = AIPS_0 (PBRIDGE_B)</p> |

**12.1.2 Bus Bridge Configuration Register 1 (PCM\_IAHB\_BE1)**

Address: 0h base + 4h offset = 4h

|       |    |    |    |    |    |             |             |             |    |    |    |    |    |             |             |             |
|-------|----|----|----|----|----|-------------|-------------|-------------|----|----|----|----|----|-------------|-------------|-------------|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5           | 6           | 7           | 8  | 9  | 10 | 11 | 12 | 13          | 14          | 15          |
| R     |    |    | 0  |    |    |             |             |             |    |    | 0  |    |    |             |             |             |
| W     |    |    |    |    |    | PRE_CORE1_D | BRE_CORE1_D | BWE_CORE1_D |    |    |    |    |    | PRE_CORE1_I | BRE_CORE1_I | BWE_CORE1_I |
| Reset | 0  | 0  | 0  | 0  | 0  | 0           | 1           | 1           | 0  | 0  | 0  | 0  | 0  | 0           | 1           | 1           |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21          | 22          | 23          | 24 | 25 | 26 | 27 | 28 | 29          | 30          | 31          |
| R     |    |    | 0  |    |    |             |             |             |    |    | 0  |    |    |             |             |             |
| W     |    |    |    |    |    | PRE_CORE0_D | BRE_CORE0_D | BWE_CORE0_D |    |    |    |    |    | PRE_CORE0_I | BRE_CORE0_I | BWE_CORE0_I |
| Reset | 0  | 0  | 0  | 0  | 0  | 0           | 1           | 1           | 0  | 0  | 0  | 0  | 0  | 0           | 1           | 1           |

**PCM\_IAHB\_BE1 field descriptions**

| Field           | Description                                                                                        |
|-----------------|----------------------------------------------------------------------------------------------------|
| 0–4<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p> |

*Table continues on the next page...*

**PCM\_IAHB\_BE1 field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 5<br>PRE_CORE1_D  | <p>Pending Read Enable Core1 Data</p> <p>This bit controls the bus gasket's handling of pending read transactions.</p> <p>0 Pending reads are disabled.<br/>1 Pending reads are enabled.</p>                                                                                                                                                                                                                                          |
| 6<br>BRE_CORE1_D  | <p>Burst Read Enable Core1 Data</p> <p>This bit controls the bus gasket's handling of burst read transactions.</p> <p>0 Burst reads are converted into a series of single transactions on the slave side of the gasket.<br/>1 Burst reads are optimized for best system performance.</p>                                                                                                                                              |
| 7<br>BWE_CORE1_D  | <p>Burst Write Enable Core1 Data</p> <p>This bit controls the bus gasket's handling of burst write transactions.</p> <p>0 Burst writes are converted into a series of single transactions on the slave side of the gasket.<br/>1 Burst writes are optimized for best system performance. Note this setting treats writes as "imprecise" such that an error response on any beat of the burst is reported on the last beat.</p>        |
| 8–12<br>Reserved  | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                    |
| 13<br>PRE_CORE1_I | <p>Pending Read Enable Core1 Instruction</p> <p>This bit controls the bus gasket's handling of pending read transactions.</p> <p>0 Pending reads are disabled.<br/>1 Pending reads are enabled.</p>                                                                                                                                                                                                                                   |
| 14<br>BRE_CORE1_I | <p>Burst Read Enable Core1 Instruction</p> <p>This bit controls the bus gasket's handling of burst read transactions.</p> <p>0 Burst reads are converted into a series of single transactions on the slave side of the gasket.<br/>1 Burst reads are optimized for best system performance.</p>                                                                                                                                       |
| 15<br>BWE_CORE1_I | <p>Burst Write Enable Core1 Instruction</p> <p>This bit controls the bus gasket's handling of burst write transactions.</p> <p>0 Burst writes are converted into a series of single transactions on the slave side of the gasket.<br/>1 Burst writes are optimized for best system performance. Note this setting treats writes as "imprecise" such that an error response on any beat of the burst is reported on the last beat.</p> |
| 16–20<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                    |
| 21<br>PRE_CORE0_D | <p>Pending Read Enable Core0 Data</p> <p>This bit controls the bus gasket's handling of pending read transactions.</p> <p>0 Pending reads are disabled<br/>1 Pending reads are enabled.</p>                                                                                                                                                                                                                                           |
| 22<br>BRE_CORE0_D | <p>Burst Read Enable Core0 Data</p> <p>This bit controls the bus gasket's handling of burst read transactions.</p>                                                                                                                                                                                                                                                                                                                    |

*Table continues on the next page...*

**PCM\_IAHB\_BE1 field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | 0 Burst reads are converted into a series of single transactions on the slave side of the gasket.<br>1 Burst reads are optimized for best system performance.                                                                                                                                                                                                                                                                 |
| 23<br>BWE_CORE0_D | Burst Write Enable Core0 Data<br><br>This bit controls the bus gasket's handling of burst write transactions.<br><br>0 Burst writes are converted into a series of single transactions on the slave side of the gasket.<br>1 Burst writes are optimized for best system performance. Note this setting treats writes as "imprecise" such that an error response on any beat of the burst is reported on the last beat.        |
| 24–28<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                       |
| 29<br>PRE_CORE0_I | Pending Read Enable Core0 Instruction<br><br>This bit controls the bus gasket's handling of pending read transactions.<br><br>0 Pending reads are disabled.<br>1 Pending reads are enabled.                                                                                                                                                                                                                                   |
| 30<br>BRE_CORE0_I | Burst Read Enable Core0 Instruction<br><br>This bit controls the bus gasket's handling of burst read transactions.<br><br>0 Burst reads are converted into a series of single transactions on the slave side of the gasket.<br>1 Burst reads are optimized for best system performance.                                                                                                                                       |
| 31<br>BWE_CORE0_I | Burst Write Enable Core0 Instruction<br><br>This bit controls the bus gasket's handling of burst write transactions.<br><br>0 Burst writes are converted into a series of single transactions on the slave side of the gasket.<br>1 Burst writes are optimized for best system performance. Note this setting treats writes as "imprecise" such that an error response on any beat of the burst is reported on the last beat. |

### 12.1.3 Bus Bridge Configuration Register 2 (PCM\_IAHB\_BE2)

Address: 0h base + 8h offset = 8h

|       |    |    |    |    |    |           |           |           |    |    |    |    |    |           |           |           |
|-------|----|----|----|----|----|-----------|-----------|-----------|----|----|----|----|----|-----------|-----------|-----------|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5         | 6         | 7         | 8  | 9  | 10 | 11 | 12 | 13        | 14        | 15        |
| R     | 0  |    |    |    |    | PRE_FEC   | BRE_FEC   | BWE_FEC   | 0  |    |    |    |    | PRE_M6    | BRE_M6    | BWE_M6    |
| W     |    |    |    |    |    |           |           |           |    |    |    |    |    |           |           |           |
| Reset | 0  | 0  | 0  | 0  | 0  | 0         | 1         | 1         | 0  | 0  | 0  | 0  | 0  | 0         | 1         | 1         |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21        | 22        | 23        | 24 | 25 | 26 | 27 | 28 | 29        | 30        | 31        |
| R     | 0  |    |    |    |    | PRE_DMA_B | BRE_DMA_B | BWE_DMA_B | 0  |    |    |    |    | PRE_DMA_A | BRE_DMA_A | BWE_DMA_A |
| W     |    |    |    |    |    |           |           |           |    |    |    |    |    |           |           |           |
| Reset | 0  | 0  | 0  | 0  | 0  | 0         | 1         | 1         | 0  | 0  | 0  | 0  | 0  | 0         | 1         | 1         |

#### PCM\_IAHB\_BE2 field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                     |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–4<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                         |
| 5<br>PRE_FEC     | Pending Read Enable FEC<br><br>This bit controls the bus gasket's handling of pending read transactions.<br><br>0 Pending reads are disabled.<br>1 Pending reads are enabled.                                                                                                                                                                                                                                   |
| 6<br>BRE_FEC     | Burst Read Enable FEC<br><br>This bit controls the bus gasket's handling of burst read transactions.<br><br>0 Burst reads are converted into a series of single transactions on the slave side of the gasket.<br>1 Burst reads are optimized for best system performance.                                                                                                                                       |
| 7<br>BWE_FEC     | Burst Write Enable FEC<br><br>This bit controls the bus gasket's handling of burst write transactions.<br><br>0 Burst writes are converted into a series of single transactions on the slave side of the gasket.<br>1 Burst writes are optimized for best system performance. Note this setting treats writes as "imprecise" such that an error response on any beat of the burst is reported on the last beat. |
| 8–12<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                         |
| 13<br>PRE_M6     | Pending Read Enable Master Port 6 Concentrator<br><br>This bit controls the bus gasket's handling of pending read transactions.                                                                                                                                                                                                                                                                                 |

Table continues on the next page...



**PCM\_IAHB\_BE2 field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | 0 Pending reads are disabled.<br>1 Pending reads are enabled.                                                                                                                                                                                                                                                                                                                                                                          |
| 14<br>BRE_M6      | Burst Read Enable Master Port 6 Concentrator<br><br>This bit controls the bus gasket's handling of burst read transactions.<br><br>0 Burst reads are converted into a series of single transactions on the slave side of the gasket.<br>1 Burst reads are optimized for best system performance.                                                                                                                                       |
| 15<br>BWE_M6      | Burst Write Enable Master Port 6 Concentrator<br><br>This bit controls the bus gasket's handling of burst write transactions.<br><br>0 Burst writes are converted into a series of single transactions on the slave side of the gasket.<br>1 Burst writes are optimized for best system performance. Note this setting treats writes as "imprecise" such that an error response on any beat of the burst is reported on the last beat. |
| 16–20<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                |
| 21<br>PRE_DMA_B   | Pending Read Enable eDMA_B<br><br>This bit controls the bus gasket's handling of pending read transactions.<br><br>0 Pending reads are disabled<br>1 Pending reads are enabled.                                                                                                                                                                                                                                                        |
| 22<br>BRE_DMA_B   | Burst Read Enable eDMA_B<br><br>This bit controls the bus gasket's handling of burst read transactions.<br><br>0 Burst reads are converted into a series of single transactions on the slave side of the gasket.<br>1 Burst reads are optimized for best system performance.                                                                                                                                                           |
| 23<br>BWE_DMA_B   | Burst Write Enable eDMA_B<br><br>This bit controls the bus gasket's handling of burst write transactions.<br><br>0 Burst writes are converted into a series of single transactions on the slave side of the gasket.<br>1 Burst writes are optimized for best system performance. Note this setting treats writes as "imprecise" such that an error response on any beat of the burst is reported on the last beat.                     |
| 24–28<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                |
| 29<br>PRE_DMA_A   | Pending Read Enable eDMA_A<br><br>This bit controls the bus gasket's handling of pending read transactions.<br><br>0 Pending reads are disabled.<br>1 Pending reads are enabled.                                                                                                                                                                                                                                                       |
| 30<br>BRE_DMA_A   | Burst Read Enable eDMA_A<br><br>This bit controls the bus gasket's handling of burst read transactions.<br><br>0 Burst reads are converted into a series of single transactions on the slave side of the gasket.<br>1 Burst reads are optimized for best system performance.                                                                                                                                                           |

*Table continues on the next page...*

**PCM\_IAHB\_BE2 field descriptions (continued)**

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31<br>BWE_DMA_A | <p>Burst Write Enable eDMA_A</p> <p>This bit controls the bus gasket's handling of burst write transactions.</p> <p>0 Burst writes are converted into a series of single transactions on the slave side of the gasket.</p> <p>1 Burst writes are optimized for best system performance. Note this setting treats writes as "imprecise" such that an error response on any beat of the burst is reported on the last beat.</p> |

# Chapter 13

## Semaphores (SEMA4)

### 13.1 Introduction

The module provides the hardware support needed in multi-core systems for implementing semaphores and provide a simple mechanism to achieve "lock/unlock" operations via a single write access. This approach eliminates architecture-specific implementations like atomic (indivisible) read-modify-write instructions or reservation mechanisms. The result is an architecture-neutral solution that provides hardware-enforced gates as well as other useful system functions related to the gating mechanisms.

#### 13.1.1 Multi-Core Programming 101: Software Gates

Multi-processor systems require a function that can be used to safely and easily provide a locking mechanism which is then used by system software to control access to shared data structures, shared hardware resources, etc. These "gating" mechanisms are used by the software to serialize (and synchronize) writes to shared data and/or resources to prevent race conditions and preserve memory coherency between different processes and processors.

Consider the following description of a typical use-case. Processor X enters a section of code where shared data values are to be updated. It must first acquire a semaphore; this can be considered "locking (or closing) a software gate." Once the gate has been locked, a properly-architected software system does not allow other processes (or processors) to execute the same code segment or modify the shared data structure protected by the gate, that is, other processes/processors are locked out. Many software implementations include a *spin-wait loop* within the lock function until the locking of the gate is accomplished. Once the lock has been obtained, Processor X continues execution and updates the data values protected by the particular lock. Once the updates are complete, Processor X unlocks (or opens) the software gate, allowing other processes/processors access to the updated data values.

There are three important rules that must be followed for a correctly-implemented system solution:

1. All writes to shared data values or shared hardware resources must be protected by a "gate" variable.
2. Once a processor locks a gate, accesses to the shared data or resources by other processes/processors must be blocked. This is enforced by software conventions.
3. The processor that locks a particular gate is the only processor that can open (unlock) that gate.

Information in the hardware gate identifying the locking processor can be extremely useful for system-level debugging.

The Hennessy/Patterson text on computer architecture offers the following description of software gating.

"One of the major requirements of a shared-memory architecture multiprocessor is being able to coordinate processes that are working on a common task. Typically, a programmer will use *lock variables* to synchronize the processes.

The difficulty for the architect of a multiprocessor is to provide a mechanism to decide which processor gets the lock and to provide the operation that locks a variable. Arbitration is easy for shared-bus multiprocessors, since the bus is the only path to memory. The processor that gets the bus locks out all the other processors from memory. If the CPU and bus provide an atomic swap operation, programmers can create locks with the proper semantics. The adjective *atomic* is key, for it means that a processor can both read a location **and** set it to the locked value in the same bus operation, preventing any other processor from reading or writing memory." [Hennessy/Patterson, *Computer Architecture: A Quantitative Approach*, ppg. 471-472]

The classic text continues with a description of the steps required to lock/unlock a variable using an *atomic swap instruction*.

"Assume that 0 means unlocked and 1 means locked. A processor first reads the lock variable to test its state. A processor keeps reading and testing until the value indicates that the lock is unlocked. The processor then races against all other processes that were similarly "spin waiting" to see who can lock the variable first. All processes use a swap instruction that reads the old value and stores a 1 into the lock variable. The single winner will see the 0, and the losers will see a 1 that was placed there by the winner. (The losers will continue to set the variable to the locked value, but that doesn't matter.) The winning processor executes the code after the lock and then stores a 0 into the lock when it exits,

starting the race all over again. Testing the old value and then setting to a new value is why the atomic swap instruction is called *test and set* in some instruction sets."

[Hennessy/Patterson, *Computer Architecture: A Quantitative Approach*, ppg. 472-473]

For multi-core SoCs developed in the Microcontroller Division, a hardware-enforced semaphore implementation as a platform IPS module is used for the following reasons:

- A smaller module size versus a platform reservation monitor for PowerPC-based designs and it does not introduce any timing-critical paths, for example, the AHB system bus address phase cycle like a reservation monitor
- Provides a more robust implementation with features not possible in a reservation-only approach
- Provides a more generic, architecture-neutral solution, applicable across *all* the processor families supported by the Microcontroller Division

The sole drawback to a hardware-based semaphore module is the limited number of semaphores versus the "infinite" number that can be supported with PowerPC reservation instructions.

### 13.1.2 Overview

The IPS\_Semaphores module provides a platform IPS slave device which implements 16 hardware-enforced gates with the following features:

- Module definition supports 16 hardware-enforced gates in a dual-processor configuration, where cp0 is core processor 0 and cp1 is core processor 1
  - Hardware gates appear as a 16-entry byte-size array with read and write accesses
    - Processors lock gates by writing "processor\_number+1" to the appropriate gate and must read back the gate value to verify the lock operation was successful
    - Once locked, the gate is unlocked by a write of zeroes from the locking processor
- Optional interrupt notification after a failed lock write provides a mechanism to indicate when the gate is unlocked

- Secure reset mechanisms are supported to clear the contents of individual semaphore gates or notification logic, as well as a clear\_all capability
- Programming model allocates memory space to support up to 8 processors and up to 64 gates

A simplified block diagram of the Semaphores module is shown in [Figure 13-1](#). In the diagram, the register blocks named gate0, gate1,..., gate 15 include the finite state machines implementing the semaphore gates plus the interrupt notification logic.

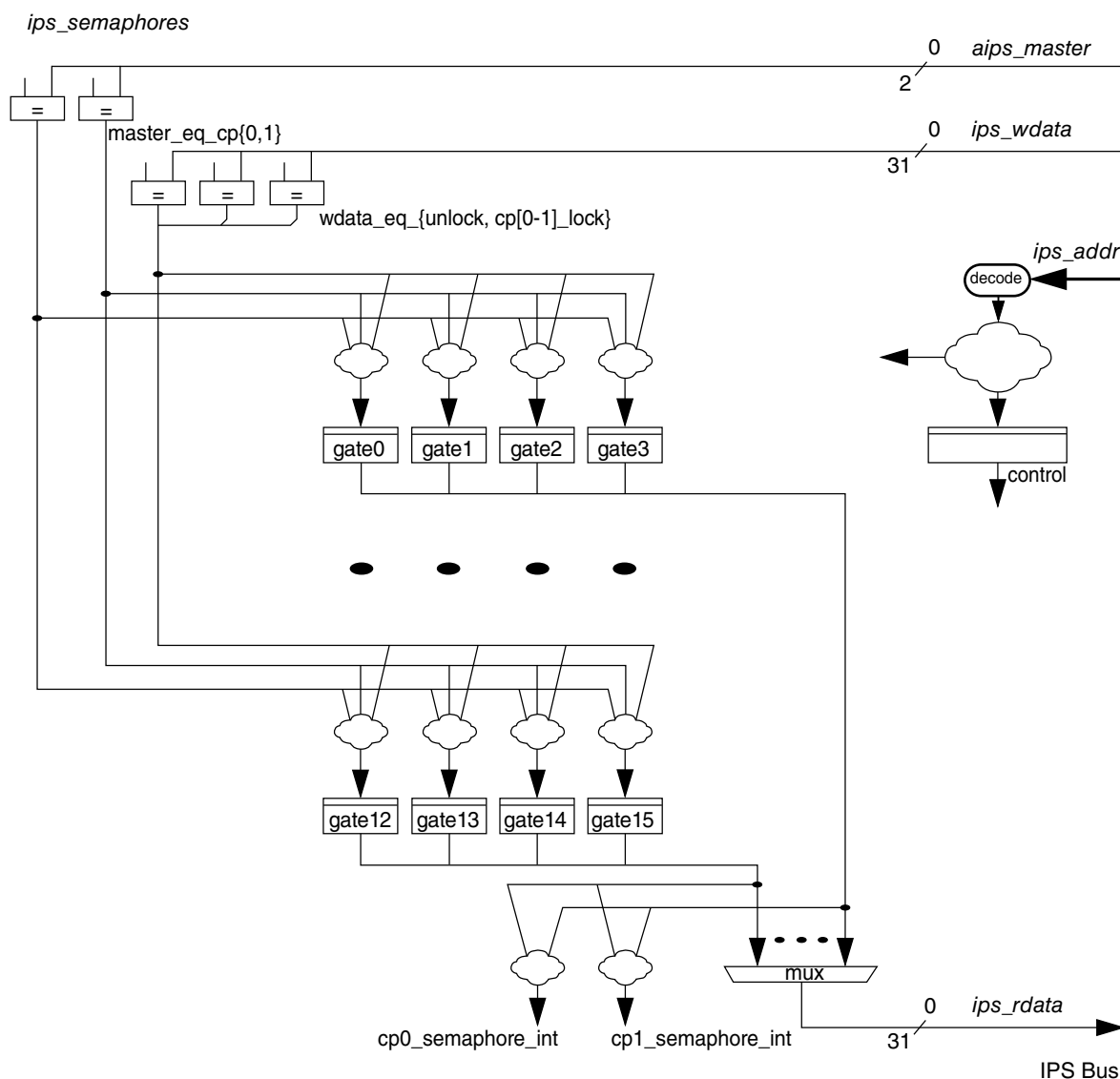


**Figure 13-1. IPS\_Semaphores Block Diagram**

### 13.1.3 Features

The Semaphores module implements hardware-enforced semaphores as an IPS-mapped slave peripheral device. The feature set includes:

- Support for 16 hardware-enforced gates in a dual-processor configuration
  - Each hardware gate appears as a 3-state, 2-bit state machine, with all 16 gates mapped as a byte-size array
    - 3-state implementation
      - if gate = 0b00, then state = unlocked
      - if gate = 0b01, then state = locked by processor 0
      - if gate = 0b10, then state = locked by processor 1
  - Uses the bus master number as a reference attribute plus the specified data patterns to validate all write operations
  - Once locked, the gate can (and must) be unlocked by a write of zeroes from the locking processor
- Optional interrupt notification after a failed lock write provides a mechanism to indicate when the gate is unlocked
- Secure reset mechanisms are supported to clear the contents of individual gates or notification logic, as well as a clear\_all capability
- Memory-mapped IPS slave peripheral platform module
  - Interface to the IPS bus for programming-model accesses
  - Two outputs (one per processor) for interrupt notification of failed lock writes

### 13.1.4 Modes of Operation

The Semaphores module does not support any special modes of operation. As a slave peripheral memory-mapped device located on the platform's IPS slave bus, it responds based strictly on the memory addresses of the connected bus. The IPS bus is used to access the Semaphores ' programming model.

## 13.2 External Signal Description

The Semaphores module does not include any external interfaces.

## 13.3 Memory map and register definition

The Semaphores module provides an IPS programming model mapped to an SPP-standard on-platform 16 KB space. The description here specifies a dual-core configuration with 16 semaphore gates. All the register names are prefixed with "Sema4" as an abbreviation for the full module name.

The programming model is referenced using 8-, 16- and 32-bit accesses. Reads can use any reference size, while writes are generally restricted to the size of the register. Exceptions to the write size restrictions are detailed in the individual register descriptions. Attempted references using inappropriate access sizes, to undefined (reserved) addresses, or with a non-supported access type (for example, a write to a read-only register) generate an IPS error termination.

Finally, the programming model allocates space for a definition with up to 64 gates and up to 8 processor cores, even though this definition is considerably larger than any currently-planned module implementations. The number of gates and supported processor cores are independent; there is no relationship between these two system variables.

The 16 KB Semaphores programming model map is shown in the following table.

**SEMA4 memory map**

| Address offset (hex) | Register name                              | Width (in bits) | Access | Reset value | Section/ page               |
|----------------------|--------------------------------------------|-----------------|--------|-------------|-----------------------------|
| 0                    | Semaphores Gate 0 Register (SEMA4_Gate00)  | 8               | R/W    | 00h         | <a href="#">13.3.1/469</a>  |
| 1                    | Semaphores Gate 1 Register (SEMA4_Gate01)  | 8               | R/W    | 00h         | <a href="#">13.3.2/470</a>  |
| 2                    | Semaphores Gate 2 Register (SEMA4_Gate02)  | 8               | R/W    | 00h         | <a href="#">13.3.3/471</a>  |
| 3                    | Semaphores Gate 3 Register (SEMA4_Gate03)  | 8               | R/W    | 00h         | <a href="#">13.3.4/472</a>  |
| 4                    | Semaphores Gate 4 Register (SEMA4_Gate04)  | 8               | R/W    | 00h         | <a href="#">13.3.5/473</a>  |
| 5                    | Semaphores Gate 5 Register (SEMA4_Gate05)  | 8               | R/W    | 00h         | <a href="#">13.3.6/474</a>  |
| 6                    | Semaphores Gate 6 Register (SEMA4_Gate06)  | 8               | R/W    | 00h         | <a href="#">13.3.7/475</a>  |
| 7                    | Semaphores Gate 7 Register (SEMA4_Gate07)  | 8               | R/W    | 00h         | <a href="#">13.3.8/476</a>  |
| 8                    | Semaphores Gate 8 Register (SEMA4_Gate08)  | 8               | R/W    | 00h         | <a href="#">13.3.9/477</a>  |
| 9                    | Semaphores Gate 9 Register (SEMA4_Gate09)  | 8               | R/W    | 00h         | <a href="#">13.3.10/478</a> |
| A                    | Semaphores Gate 10 Register (SEMA4_Gate10) | 8               | R/W    | 00h         | <a href="#">13.3.11/479</a> |
| B                    | Semaphores Gate 11 Register (SEMA4_Gate11) | 8               | R/W    | 00h         | <a href="#">13.3.12/480</a> |
| C                    | Semaphores Gate 12 Register (SEMA4_Gate12) | 8               | R/W    | 00h         | <a href="#">13.3.13/481</a> |
| D                    | Semaphores Gate 13 Register (SEMA4_Gate13) | 8               | R/W    | 00h         | <a href="#">13.3.14/482</a> |
| E                    | Semaphores Gate 14 Register (SEMA4_Gate14) | 8               | R/W    | 00h         | <a href="#">13.3.15/483</a> |

*Table continues on the next page...*



**SEMA4 memory map (continued)**

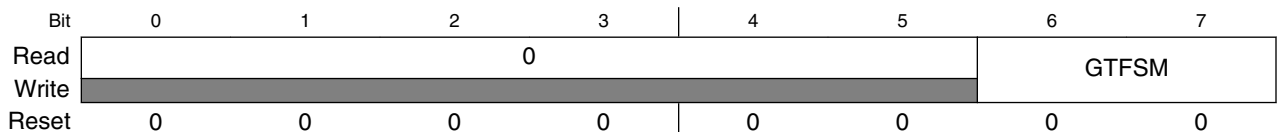
| Address offset (hex) | Register name                                                 | Width (in bits) | Access | Reset value | Section/ page               |
|----------------------|---------------------------------------------------------------|-----------------|--------|-------------|-----------------------------|
| F                    | Semaphores Gate 15 Register (SEMA4_Gate15)                    | 8               | R/W    | 00h         | <a href="#">13.3.16/484</a> |
| 40                   | Semaphores Processor n IRQ Notification Enable (SEMA4_CP0INE) | 16              | R/W    | 0000h       | <a href="#">13.3.17/485</a> |
| 48                   | Semaphores Processor n IRQ Notification Enable (SEMA4_CP1INE) | 16              | R/W    | 0000h       | <a href="#">13.3.17/485</a> |
| 80                   | Semaphores Processor n IRQ Notification (SEMA4_CP0NTF)        | 16              | R      | 0000h       | <a href="#">13.3.18/487</a> |
| 88                   | Semaphores Processor n IRQ Notification (SEMA4_CP1NTF)        | 16              | R      | 0000h       | <a href="#">13.3.18/487</a> |
| 100                  | Semaphores (Secure) Reset Gate n (SEMA4_RSTGT)                | 16              | R/W    | 0000h       | <a href="#">13.3.19/488</a> |
| 104                  | Semaphores (Secure) Reset IRQ Notification (SEMA4_RSTNTF)     | 16              | R/W    | 0000h       | <a href="#">13.3.20/490</a> |

**13.3.1 Semaphores Gate 0 Register (SEMA4\_Gate00)**

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 0h base + 0h offset = 0h

**SEMA4\_Gate00 field descriptions**

| Field           | Description                                                                             |
|-----------------|-----------------------------------------------------------------------------------------|
| 0–5<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |

*Table continues on the next page...*

**SEMA4\_Gate00 field descriptions (continued)**

| Field        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 6–7<br>GTFSM | <p>Gate Finite State Machine.</p> <p>Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see <a href="#">SEMA4_GATEn Operation</a> .</p> <p><b>NOTE:</b> The state of the gate reflects the last processor that locked it, which can be useful during system debug.</p> <p>00 The gate is unlocked (free).<br/> 01 The gate has been locked by processor 0.<br/> 10 The gate has been locked by processor 1.<br/> 11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine.</p> |

**13.3.2 Semaphores Gate 1 Register (SEMA4\_Gate01)**

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 0h base + 1h offset = 1h

|       |   |   |   |   |   |       |   |   |
|-------|---|---|---|---|---|-------|---|---|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5     | 6 | 7 |
| Read  | 0 |   |   |   |   | GTFSM |   |   |
| Write |   |   |   |   |   |       |   |   |
| Reset | 0 | 0 | 0 | 0 | 0 | 0     | 0 | 0 |

**SEMA4\_Gate01 field descriptions**

| Field           | Description                                                                                                                                                                                                                                             |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–5<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                      |
| 6–7<br>GTFSM    | <p>Gate Finite State Machine.</p> <p>Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see <a href="#">SEMA4_GATEn Operation</a> .</p> |

*Table continues on the next page...*

**SEMA4\_Gate01 field descriptions (continued)**

| Field | Description                                                                                                                                                |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | <b>NOTE:</b> The state of the gate reflects the last processor that locked it, which can be useful during system debug.                                    |
| 00    | The gate is unlocked (free).                                                                                                                               |
| 01    | The gate has been locked by processor 0.                                                                                                                   |
| 10    | The gate has been locked by processor 1.                                                                                                                   |
| 11    | This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine. |

**13.3.3 Semaphores Gate 2 Register (SEMA4\_Gate02)**

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 0h base + 2h offset = 2h

|       |   |   |   |   |   |       |   |   |
|-------|---|---|---|---|---|-------|---|---|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5     | 6 | 7 |
| Read  | 0 |   |   |   |   | GTFSM |   |   |
| Write |   |   |   |   |   |       |   |   |
| Reset | 0 | 0 | 0 | 0 | 0 | 0     | 0 | 0 |

**SEMA4\_Gate02 field descriptions**

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–5<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                |
| 6–7<br>GTFSM    | Gate Finite State Machine.<br><br>Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see <a href="#">SEMA4_GATEn Operation</a> .<br><br><b>NOTE:</b> The state of the gate reflects the last processor that locked it, which can be useful during system debug.<br><br>00 The gate is unlocked (free). |

*Table continues on the next page...*

**SEMA4\_Gate02 field descriptions (continued)**

| Field | Description                                                                                                                                                |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 01    | The gate has been locked by processor 0.                                                                                                                   |
| 10    | The gate has been locked by processor 1.                                                                                                                   |
| 11    | This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine. |

**13.3.4 Semaphores Gate 3 Register (SEMA4\_Gate03)**

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 0h base + 3h offset = 3h

|       |   |   |   |   |   |       |   |   |
|-------|---|---|---|---|---|-------|---|---|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5     | 6 | 7 |
| Read  | 0 |   |   |   |   | GTFSM |   |   |
| Write |   |   |   |   |   |       |   |   |
| Reset | 0 | 0 | 0 | 0 | 0 | 0     | 0 | 0 |

**SEMA4\_Gate03 field descriptions**

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–5<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 6–7<br>GTFSM    | Gate Finite State Machine.<br><br>Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see <a href="#">SEMA4_GATEn Operation</a> .<br><br><b>NOTE:</b> The state of the gate reflects the last processor that locked it, which can be useful during system debug.<br><br>00 The gate is unlocked (free).<br>01 The gate has been locked by processor 0.<br>10 The gate has been locked by processor 1.<br>11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine. |

### 13.3.5 Semaphores Gate 4 Register (SEMA4\_Gate04)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 0h base + 4h offset = 4h

|       |   |   |   |   |   |       |   |   |
|-------|---|---|---|---|---|-------|---|---|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5     | 6 | 7 |
| Read  | 0 |   |   |   |   | GTFSM |   |   |
| Write |   |   |   |   |   |       |   |   |
| Reset | 0 | 0 | 0 | 0 | 0 | 0     | 0 | 0 |

**SEMA4\_Gate04 field descriptions**

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–5<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 6–7<br>GTFSM    | Gate Finite State Machine.<br><br>Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see <a href="#">SEMA4_GATEn Operation</a> .<br><br><b>NOTE:</b> The state of the gate reflects the last processor that locked it, which can be useful during system debug.<br><br>00 The gate is unlocked (free).<br>01 The gate has been locked by processor 0.<br>10 The gate has been locked by processor 1.<br>11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine. |

13.3.6 Semaphores Gate 5 Register (SEMA4\_Gate05)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 0h base + 5h offset = 5h



SEMA4\_Gate05 field descriptions

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–5<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 6–7<br>GTFSM    | <p>Gate Finite State Machine.</p> <p>Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see <a href="#">SEMA4_GATEn Operation</a> .</p> <p><b>NOTE:</b> The state of the gate reflects the last processor that locked it, which can be useful during system debug.</p> <p>00 The gate is unlocked (free).</p> <p>01 The gate has been locked by processor 0.</p> <p>10 The gate has been locked by processor 1.</p> <p>11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine.</p> |

### 13.3.7 Semaphores Gate 6 Register (SEMA4\_Gate06)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 0h base + 6h offset = 6h

|       |   |   |   |   |   |       |   |   |
|-------|---|---|---|---|---|-------|---|---|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5     | 6 | 7 |
| Read  | 0 |   |   |   |   | GTFSM |   |   |
| Write |   |   |   |   |   |       |   |   |
| Reset | 0 | 0 | 0 | 0 | 0 | 0     | 0 | 0 |

**SEMA4\_Gate06 field descriptions**

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–5<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 6–7<br>GTFSM    | Gate Finite State Machine.<br><br>Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see <a href="#">SEMA4_GATEn Operation</a> .<br><br><b>NOTE:</b> The state of the gate reflects the last processor that locked it, which can be useful during system debug.<br><br>00 The gate is unlocked (free).<br>01 The gate has been locked by processor 0.<br>10 The gate has been locked by processor 1.<br>11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine. |

13.3.8 Semaphores Gate 7 Register (SEMA4\_Gate07)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 0h base + 7h offset = 7h



SEMA4\_Gate07 field descriptions

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–5<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 6–7<br>GTFSM    | <p>Gate Finite State Machine.</p> <p>Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see <a href="#">SEMA4_GATEn Operation</a> .</p> <p><b>NOTE:</b> The state of the gate reflects the last processor that locked it, which can be useful during system debug.</p> <p>00 The gate is unlocked (free).</p> <p>01 The gate has been locked by processor 0.</p> <p>10 The gate has been locked by processor 1.</p> <p>11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine.</p> |



### 13.3.9 Semaphores Gate 8 Register (SEMA4\_Gate08)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 0h base + 8h offset = 8h

|       |   |   |   |   |   |       |   |   |
|-------|---|---|---|---|---|-------|---|---|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5     | 6 | 7 |
| Read  | 0 |   |   |   |   | GTFSM |   |   |
| Write |   |   |   |   |   |       |   |   |
| Reset | 0 | 0 | 0 | 0 | 0 | 0     | 0 | 0 |

**SEMA4\_Gate08 field descriptions**

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–5<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 6–7<br>GTFSM    | Gate Finite State Machine.<br><br>Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see <a href="#">SEMA4_GATEn Operation</a> .<br><br><b>NOTE:</b> The state of the gate reflects the last processor that locked it, which can be useful during system debug.<br><br>00 The gate is unlocked (free).<br>01 The gate has been locked by processor 0.<br>10 The gate has been locked by processor 1.<br>11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine. |

13.3.10 Semaphores Gate 9 Register (SEMA4\_Gate09)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 0h base + 9h offset = 9h



SEMA4\_Gate09 field descriptions

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–5<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 6–7<br>GTFSM    | Gate Finite State Machine.<br><br>Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see <a href="#">SEMA4_GATEn Operation</a> .<br><br><b>NOTE:</b> The state of the gate reflects the last processor that locked it, which can be useful during system debug.<br><br>00 The gate is unlocked (free).<br>01 The gate has been locked by processor 0.<br>10 The gate has been locked by processor 1.<br>11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine. |

### 13.3.11 Semaphores Gate 10 Register (SEMA4\_Gate10)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 0h base + Ah offset = Ah

|       |   |   |   |   |   |       |   |   |
|-------|---|---|---|---|---|-------|---|---|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5     | 6 | 7 |
| Read  | 0 |   |   |   |   | GTFSM |   |   |
| Write |   |   |   |   |   |       |   |   |
| Reset | 0 | 0 | 0 | 0 | 0 | 0     | 0 | 0 |

**SEMA4\_Gate10 field descriptions**

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–5<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 6–7<br>GTFSM    | Gate Finite State Machine.<br><br>Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see <a href="#">SEMA4_GATEn Operation</a> .<br><br><b>NOTE:</b> The state of the gate reflects the last processor that locked it, which can be useful during system debug.<br><br>00 The gate is unlocked (free).<br>01 The gate has been locked by processor 0.<br>10 The gate has been locked by processor 1.<br>11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine. |

13.3.12 Semaphores Gate 11 Register (SEMA4\_Gate11)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 0h base + Bh offset = Bh



SEMA4\_Gate11 field descriptions

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–5<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 6–7<br>GTFSM    | Gate Finite State Machine.<br><br>Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see <a href="#">SEMA4_GATEn Operation</a> .<br><br><b>NOTE:</b> The state of the gate reflects the last processor that locked it, which can be useful during system debug.<br><br>00 The gate is unlocked (free).<br>01 The gate has been locked by processor 0.<br>10 The gate has been locked by processor 1.<br>11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine. |

### 13.3.13 Semaphores Gate 12 Register (SEMA4\_Gate12)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 0h base + Ch offset = Ch

|       |   |   |   |   |   |       |   |   |
|-------|---|---|---|---|---|-------|---|---|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5     | 6 | 7 |
| Read  | 0 |   |   |   |   | GTFSM |   |   |
| Write |   |   |   |   |   |       |   |   |
| Reset | 0 | 0 | 0 | 0 | 0 | 0     | 0 | 0 |

**SEMA4\_Gate12 field descriptions**

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–5<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 6–7<br>GTFSM    | Gate Finite State Machine.<br><br>Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see <a href="#">SEMA4_GATEn Operation</a> .<br><br><b>NOTE:</b> The state of the gate reflects the last processor that locked it, which can be useful during system debug.<br><br>00 The gate is unlocked (free).<br>01 The gate has been locked by processor 0.<br>10 The gate has been locked by processor 1.<br>11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine. |

13.3.14 Semaphores Gate 13 Register (SEMA4\_Gate13)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 0h base + Dh offset = Dh



SEMA4\_Gate13 field descriptions

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–5<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 6–7<br>GTFSM    | Gate Finite State Machine.<br><br>Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see <a href="#">SEMA4_GATEn Operation</a> .<br><br><b>NOTE:</b> The state of the gate reflects the last processor that locked it, which can be useful during system debug.<br><br>00 The gate is unlocked (free).<br>01 The gate has been locked by processor 0.<br>10 The gate has been locked by processor 1.<br>11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine. |

### 13.3.15 Semaphores Gate 14 Register (SEMA4\_Gate14)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 0h base + Eh offset = Eh

|       |   |   |   |   |   |       |   |   |
|-------|---|---|---|---|---|-------|---|---|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5     | 6 | 7 |
| Read  | 0 |   |   |   |   | GTFSM |   |   |
| Write |   |   |   |   |   |       |   |   |
| Reset | 0 | 0 | 0 | 0 | 0 | 0     | 0 | 0 |

**SEMA4\_Gate14 field descriptions**

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–5<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 6–7<br>GTFSM    | Gate Finite State Machine.<br><br>Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see <a href="#">SEMA4_GATEn Operation</a> .<br><br><b>NOTE:</b> The state of the gate reflects the last processor that locked it, which can be useful during system debug.<br><br>00 The gate is unlocked (free).<br>01 The gate has been locked by processor 0.<br>10 The gate has been locked by processor 1.<br>11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine. |

13.3.16 Semaphores Gate 15 Register (SEMA4\_Gate15)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 0h base + Fh offset = Fh



SEMA4\_Gate15 field descriptions

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–5<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 6–7<br>GTFSM    | <p>Gate Finite State Machine.</p> <p>Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see <a href="#">SEMA4_GATEn Operation</a> .</p> <p><b>NOTE:</b> The state of the gate reflects the last processor that locked it, which can be useful during system debug.</p> <p>00 The gate is unlocked (free).</p> <p>01 The gate has been locked by processor 0.</p> <p>10 The gate has been locked by processor 1.</p> <p>11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine.</p> |



### 13.3.17 Semaphores Processor n IRQ Notification Enable (SEMA4\_CPnINE)

The application of a hardware semaphore module provides an opportunity for implementation of helpful system-level features. An example is an optional mechanism to generate a processor interrupt after a failed lock attempt. Recall traditional software gate functions execute a spin-wait loop in an effort to obtain and lock the referenced gate. With this module, the processor that fails in the lock attempt could continue with other tasks and allow a properly-enabled notification interrupt to return its execution to the original lock function.

The optional notification interrupt function consists of two registers for each processor: an interrupt notification enable register (SEMA4\_CPnINE) and the interrupt request register (SEMA4\_CPnNTF). To support implementations with more than 16 gates, these registers can be referenced with aligned 16- or 32-bit accesses. For the SEMA4\_CPnINE registers, unimplemented bits read as zeroes, and writes are ignored.

Address: 0h base + 40h offset + (8d × i), where i=0d to 1d

|       |      |      |      |      |      |      |      |      |
|-------|------|------|------|------|------|------|------|------|
| Bit   | 0    | 1    | 2    | 3    | 4    | 5    | 6    | 7    |
| Read  | INE0 | INE1 | INE2 | INE3 | INE4 | INE5 | INE6 | INE7 |
| Write |      |      |      |      |      |      |      |      |
| Reset | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |

|       |      |      |       |       |       |       |       |       |
|-------|------|------|-------|-------|-------|-------|-------|-------|
| Bit   | 8    | 9    | 10    | 11    | 12    | 13    | 14    | 15    |
| Read  | INE8 | INE9 | INE10 | INE11 | INE12 | INE13 | INE14 | INE15 |
| Write |      |      |       |       |       |       |       |       |
| Reset | 0    | 0    | 0     | 0     | 0     | 0     | 0     | 0     |

#### SEMA4\_CPnINE field descriptions

| Field     | Description                                                                                                                                                                                                                                                                                  |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>INE0 | Interrupt Request Notification Enable 0. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 0.<br><br>0 The generation of the notification interrupt is disabled.<br>1 The generation of the notification interrupt is enabled. |
| 1<br>INE1 | Interrupt Request Notification Enable 1. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 1.<br><br>0 The generation of the notification interrupt is disabled.<br>1 The generation of the notification interrupt is enabled. |
| 2<br>INE2 | Interrupt Request Notification Enable 2. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 2.<br><br>0 The generation of the notification interrupt is disabled.<br>1 The generation of the notification interrupt is enabled. |
| 3<br>INE3 | Interrupt Request Notification Enable 3. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 3.                                                                                                                                  |

*Table continues on the next page...*

**SEMA4\_CPnINE field descriptions (continued)**

| Field       | Description                                                                                                                                                                                                                                                                                    |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|             | 0 The generation of the notification interrupt is disabled.<br>1 The generation of the notification interrupt is enabled.                                                                                                                                                                      |
| 4<br>INE4   | Interrupt Request Notification Enable 4. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 4.<br><br>0 The generation of the notification interrupt is disabled.<br>1 The generation of the notification interrupt is enabled.   |
| 5<br>INE5   | Interrupt Request Notification Enable 5. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 5.<br><br>0 The generation of the notification interrupt is disabled.<br>1 The generation of the notification interrupt is enabled.   |
| 6<br>INE6   | Interrupt Request Notification Enable 6. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 6.<br><br>0 The generation of the notification interrupt is disabled.<br>1 The generation of the notification interrupt is enabled.   |
| 7<br>INE7   | Interrupt Request Notification Enable 7. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 7.<br><br>0 The generation of the notification interrupt is disabled.<br>1 The generation of the notification interrupt is enabled.   |
| 8<br>INE8   | Interrupt Request Notification Enable 8. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 8.<br><br>0 The generation of the notification interrupt is disabled.<br>1 The generation of the notification interrupt is enabled.   |
| 9<br>INE9   | Interrupt Request Notification Enable 9. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 9.<br><br>0 The generation of the notification interrupt is disabled.<br>1 The generation of the notification interrupt is enabled.   |
| 10<br>INE10 | Interrupt Request Notification Enable 10. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 10.<br><br>0 The generation of the notification interrupt is disabled.<br>1 The generation of the notification interrupt is enabled. |
| 11<br>INE11 | Interrupt Request Notification Enable 11. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 11.<br><br>0 The generation of the notification interrupt is disabled.<br>1 The generation of the notification interrupt is enabled. |
| 12<br>INE12 | Interrupt Request Notification Enable 12. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 12.<br><br>0 The generation of the notification interrupt is disabled.<br>1 The generation of the notification interrupt is enabled. |
| 13<br>INE13 | Interrupt Request Notification Enable 13. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 13.                                                                                                                                  |

*Table continues on the next page...*

**SEMA4\_CPnINE field descriptions (continued)**

| Field       | Description                                                                                                                                                                                                                                                                                    |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|             | 0 The generation of the notification interrupt is disabled.<br>1 The generation of the notification interrupt is enabled.                                                                                                                                                                      |
| 14<br>INE14 | Interrupt Request Notification Enable 14. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 14.<br><br>0 The generation of the notification interrupt is disabled.<br>1 The generation of the notification interrupt is enabled. |
| 15<br>INE15 | Interrupt Request Notification Enable 15. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 15.<br><br>0 The generation of the notification interrupt is disabled.<br>1 The generation of the notification interrupt is enabled. |

**13.3.18 Semaphores Processor n IRQ Notification (SEMA4\_CPnNTF)**

The Semaphores module optionally allows the processor that fails in the lock attempt to continue with other tasks and allow a properly-enabled notification interrupt to return its execution to the original lock function rather than simply execute in a spin-wait loop.

The optional notification interrupt mechanism consists of two registers for each processor: an interrupt notification enable register (SEMA4\_CPnINE) and the read-only notification interrupt request register (SEMA4\_CPnNTF). To support implementations with more than 16 gates, these registers can be referenced with aligned 16- or 32-bit accesses. For the SEMA4\_CPnNTF registers, unimplemented bits read as zeroes.

The notification interrupt is generated via a unique finite state machine, one per hardware gate. This machine operates in the following manner:

1. When an attempted lock fails, the FSM enters a first state where it waits until the gate is unlocked.
2. Once unlocked, the FSM enters a second state where it generates an interrupt request to the “failed lock” processor.
3. When the “failed lock” processor succeeds in locking the gate, the IRQ is automatically negated and the FSM returns to the idle state. However, if the other processor again locks the gate, the FSM returns to the first state, negates the interrupt request, and then waits for the gate to be unlocked (again).

The notification interrupt request is implemented in a 3-bit, 5-state machine, where two specific states are encoded and program-visible as SEMA4\_CP0NTF[GNn] and SEMA4\_CP1NTF[GNn].

## Memory map and register definition

Address: 0h base + 80h offset + (8d × i), where i=0d to 1d

| Bit   | 0   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|-------|-----|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Read  | GNn |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Write |     |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

### SEMA4\_CPnNTF field descriptions

| Field       | Description                                                                                                                                                                                      |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>GNn | Gate n Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate n. For more details, see <a href="#">SEMA4_CPnNTF Operation</a> . |

## 13.3.19 Semaphores (Secure) Reset Gate n (SEMA4\_RSTGT)

Although the intent of the hardware gate implementation specifies a protocol where the locking processor must unlock the gate, it is recognized that system operation may require a reset function to re-initialize the state of any gate(s) without requiring a system-level reset.

To support this special gate reset requirement, the Semaphores module implements a "secure" reset mechanism which allows a hardware gate (or all the gates) to be initialized by following a specific dual-write access pattern. Using a technique similar to that required for the servicing of a software watchdog timer, the secure gate reset requires two consecutive writes with predefined data patterns from the same processor to force the clearing of the specified gate(s). The required access pattern is:

1. A processor performs a 16-bit write to the SEMA4\_RSTGT memory location. The most significant byte (SEMA4\_RSTGT[RSTGDP]) must be 0xE2; the least significant byte is a "don't\_care" for this reference.
2. The same processor then performs a second 16-bit write to the SEMA4\_RSTGT location. For this write, the upper byte (SEMA4\_RSTGT[RSTGDP]) is the logical complement of the first data pattern (0x1D) and the lower byte (SEMA4\_RSTGT[RSTGTN]) specifies the gate(s) to be reset. This gate field can specify a single gate be cleared, or that all gates are cleared.
3. Reads of the SEMA4\_RSTGT location return information on the 2-bit state machine (SEMA4\_RSTGT[RSTGSM]) which implements this function, the bus master performing the reset (SEMA4\_RSTGT[RSTGMS]) and the gate number(s) last cleared (SEMA4\_RSTGT[RSTGTN]). Reads of the SEMA4\_RSTGT register do not affect the secure reset finite state machine in any manner.

Address: 0h base + 100h offset = 100h

| Bit   | 0                    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8      | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|-------|----------------------|---|---|---|---|---|---|---|--------|---|----|----|----|----|----|----|
| Read  | RSTGSM_RSTGMS_RSTGDP |   |   |   |   |   |   |   | RSTGTN |   |    |    |    |    |    |    |
| Write |                      |   |   |   |   |   |   |   |        |   |    |    |    |    |    |    |
| Reset | 0                    | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0      | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

## SEMA4\_RSTGT field descriptions

| Field                               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |           |             |           |                 |                           |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |               |                           |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |            |               |                                                                                                                                                                                                                                                 |
|-------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|-------------|-----------|-----------------|---------------------------|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|---------------------------|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–7<br>RSTGSM_<br>RSTGMS_<br>RSTGDP | <p><b>NOTE:</b> This field contains subfields that vary depending on whether it is being read or written. Sub-fields indicated as having read access are valid only for read operations. Sub-fields indicated as having write access are valid only for write operations. Bit numbering in the descriptions begins with the most significant bit numbered 0. See the following table for details.</p> <table><tr><th>Access</th><th>Sub-Field</th><th>Description</th></tr><tr><td rowspan="4">Read-Only</td><td>0-1<br/>Reserved</td><td>Reserved. Always reads 0.</td></tr><tr><td>2-3<br/>RSTGSM</td><td>Reset Gate Finite State Machine. Reads of the SEMA4_RSTGT register return the encoded state machine value. The reset state machine is maintained in a 2-bit, 3-state implementation, defined as:<br/>00 Idle, waiting for the first data pattern write.<br/>01 Waiting for the second data pattern write.<br/>10 The 2-write sequence has completed. Generate the specified gate reset(s). After the reset is performed, this machine returns to the idle (waiting for first data pattern write) state. Note that the RSTGSM = 0b10 state is valid for only a single machine cycle, so it is impossible for a read to return this value<br/>11 This state encoding is never used and therefore reserved.</td></tr><tr><td>4<br/>Reserved</td><td>Reserved. Always reads 0.</td></tr><tr><td>5-7<br/>RSTGMS</td><td>Reset Gate Bus Master. This 3-bit read-only field records the logical number of the bus master performing the gate reset function. The reset function requires that the two consecutive writes to this register be initiated by the same bus master to succeed. This field is updated each time a write to this register occurs. The association between system bus master port numbers, the associated bus master device and the logical processor number is SoC-specific. See the chip configuration chapter for this information.</td></tr><tr><td>Write-Only</td><td>0-7<br/>RSTGDP</td><td>Reset Gate Data Pattern. This write-only field is accessed with the specified data patterns on the two consecutive writes to enable the gate reset mechanism. For the first write, RSTGDP = 0xE2 while the second write requires RSTGDP = 0x1D.</td></tr></table> | Access                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Sub-Field | Description | Read-Only | 0-1<br>Reserved | Reserved. Always reads 0. | 2-3<br>RSTGSM | Reset Gate Finite State Machine. Reads of the SEMA4_RSTGT register return the encoded state machine value. The reset state machine is maintained in a 2-bit, 3-state implementation, defined as:<br>00 Idle, waiting for the first data pattern write.<br>01 Waiting for the second data pattern write.<br>10 The 2-write sequence has completed. Generate the specified gate reset(s). After the reset is performed, this machine returns to the idle (waiting for first data pattern write) state. Note that the RSTGSM = 0b10 state is valid for only a single machine cycle, so it is impossible for a read to return this value<br>11 This state encoding is never used and therefore reserved. | 4<br>Reserved | Reserved. Always reads 0. | 5-7<br>RSTGMS | Reset Gate Bus Master. This 3-bit read-only field records the logical number of the bus master performing the gate reset function. The reset function requires that the two consecutive writes to this register be initiated by the same bus master to succeed. This field is updated each time a write to this register occurs. The association between system bus master port numbers, the associated bus master device and the logical processor number is SoC-specific. See the chip configuration chapter for this information. | Write-Only | 0-7<br>RSTGDP | Reset Gate Data Pattern. This write-only field is accessed with the specified data patterns on the two consecutive writes to enable the gate reset mechanism. For the first write, RSTGDP = 0xE2 while the second write requires RSTGDP = 0x1D. |
| Access                              | Sub-Field                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |           |             |           |                 |                           |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |               |                           |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |            |               |                                                                                                                                                                                                                                                 |
| Read-Only                           | 0-1<br>Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Reserved. Always reads 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |           |             |           |                 |                           |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |               |                           |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |            |               |                                                                                                                                                                                                                                                 |
|                                     | 2-3<br>RSTGSM                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Reset Gate Finite State Machine. Reads of the SEMA4_RSTGT register return the encoded state machine value. The reset state machine is maintained in a 2-bit, 3-state implementation, defined as:<br>00 Idle, waiting for the first data pattern write.<br>01 Waiting for the second data pattern write.<br>10 The 2-write sequence has completed. Generate the specified gate reset(s). After the reset is performed, this machine returns to the idle (waiting for first data pattern write) state. Note that the RSTGSM = 0b10 state is valid for only a single machine cycle, so it is impossible for a read to return this value<br>11 This state encoding is never used and therefore reserved. |           |             |           |                 |                           |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |               |                           |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |            |               |                                                                                                                                                                                                                                                 |
|                                     | 4<br>Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Reserved. Always reads 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |           |             |           |                 |                           |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |               |                           |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |            |               |                                                                                                                                                                                                                                                 |
|                                     | 5-7<br>RSTGMS                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Reset Gate Bus Master. This 3-bit read-only field records the logical number of the bus master performing the gate reset function. The reset function requires that the two consecutive writes to this register be initiated by the same bus master to succeed. This field is updated each time a write to this register occurs. The association between system bus master port numbers, the associated bus master device and the logical processor number is SoC-specific. See the chip configuration chapter for this information.                                                                                                                                                                 |           |             |           |                 |                           |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |               |                           |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |            |               |                                                                                                                                                                                                                                                 |
| Write-Only                          | 0-7<br>RSTGDP                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Reset Gate Data Pattern. This write-only field is accessed with the specified data patterns on the two consecutive writes to enable the gate reset mechanism. For the first write, RSTGDP = 0xE2 while the second write requires RSTGDP = 0x1D.                                                                                                                                                                                                                                                                                                                                                                                                                                                      |           |             |           |                 |                           |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |               |                           |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |            |               |                                                                                                                                                                                                                                                 |
| 8–15<br>RSTGTN                      | <p>Reset Gate Number. This 8-bit field specifies the specific hardware gate to be reset. This field is updated by the second write.</p> <p>If RSTGTN &lt; 64, then reset the single gate defined by RSTGTN, else reset all the gates. The corresponding secure IRQ notification state machine(s) are also reset.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |           |             |           |                 |                           |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |               |                           |               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |            |               |                                                                                                                                                                                                                                                 |

### 13.3.20 Semaphores (Secure) Reset IRQ Notification (SEMA4\_RSTNTF)

As with the case of the secure reset function and the hardware gates, it is recognized that system operation may require a reset function to re-initialize the state of the IRQ notification logic without requiring a system-level reset.

To support this special notification reset requirement, the Semaphores module implements a "secure" reset mechanism which allows an IRQ notification (or all the notifications) to be initialized by following a specific dual-write access pattern. When successful, the specified IRQ notification state machine(s) are reset. Using a technique similar to that required for the servicing of a software watchdog timer, the secure reset mechanism requires two consecutive writes with predefined data patterns from the same processor to force the clearing of the IRQ notification(s). The required access pattern is:

1. A processor performs a 16-bit write to the SEMA4\_RSTNTF memory location. The most significant byte (SEMA4\_RSTNTF[RSTNDP]) must be 0x47; the least significant byte is a "don't\_care" for this reference.
2. The same processor then performs a second 16-bit write to the SEMA4\_RSTNTF location. For this write, the upper byte (SEMA4\_RSTNTF[RSTNDP]) is the logical complement of the first data pattern (0xB8) and the lower byte (SEMA4\_RSTNTF[RSTNTN]) specifies the notification(s) to be reset. This field can specify a single notification be cleared, or that all notifications are cleared.
3. Reads of the SEMA4\_RSTNTF location return information on the 2-bit state machine (SEMA4\_RSTNTF[RSTNSM]) which implements this function, the bus master performing the reset (SEMA4\_RSTNTF[RSTNMS]) and the notification number(s) last cleared (SEMA4\_RSTNTF[RSTNTN]). Reads of the SEMA4\_RSTNTF register do not affect the secure reset finite state machine in any manner.

Address: 0h base + 104h offset = 104h

| Bit   | 0                    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8      | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|-------|----------------------|---|---|---|---|---|---|---|--------|---|----|----|----|----|----|----|
| Read  | RSTNSM_RSTNMS_RSTNDP |   |   |   |   |   |   |   | RSTNTN |   |    |    |    |    |    |    |
| Write |                      |   |   |   |   |   |   |   |        |   |    |    |    |    |    |    |
| Reset | 0                    | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0      | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

#### SEMA4\_RSTNTF field descriptions

| Field                               | Description                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0-7<br>RSTNSM_<br>RSTNMS_<br>RSTNDP | <b>NOTE:</b> This field contains subfields that vary depending on whether it is being read or written. Sub-fields indicated as having read access are valid only for read operations. Sub-fields indicated as having write access are valid only for write operations. Bit numbering in the descriptions begins with the most significant bit numbered 0. See the following table for details. |

Table continues on the next page...

**SEMA4\_RSTNTF field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                                          |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | <b>Access</b>                                                                                                                                                                                                                                                                        | <b>Sub-Field</b> | <b>Description</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|                | Read-Only                                                                                                                                                                                                                                                                            | 0-1<br>Reserved  | Reserved. Always reads 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|                |                                                                                                                                                                                                                                                                                      | 2-3<br>RSTNSM    | Reset Notification Finite State Machine. Reads of the SEMA4_RSTNTF register return the encoded state machine value. The reset state machine is maintained in a 2-bit, 3-state implementation, defined as:<br>00 Idle, waiting for the first data pattern write.<br>01 Waiting for the second data pattern write.<br>10 The 2-write sequence has completed. Generate the specified notification reset(s). After the reset is performed, this machine returns to the idle (waiting for first data pattern write) state. Note the RSTNSM = 10 state is valid for only a single machine cycle, so it is impossible for a read to return this value.<br>11 This state encoding is never used and therefore reserved.. |
|                |                                                                                                                                                                                                                                                                                      | 4<br>Reserved    | Reserved. Always reads 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|                |                                                                                                                                                                                                                                                                                      | 5-7<br>RSTNMS    | Reset Notification Bus Master. This 3-bit read-only field records the logical number of the bus master performing the notification reset function. The reset function requires that the two consecutive writes to this register be initiated by the same bus master to succeed. This field is updated each time a write to this register occurs. The association between system bus master port numbers, the associated bus master device and the logical processor number is SoC-specific. See the chip configuration chapter for this information.                                                                                                                                                             |
|                |                                                                                                                                                                                                                                                                                      | 0-7<br>RSTNDP    | Reset Notification Data Pattern. This write-only field is accessed with the specified data patterns on the two consecutive writes to enable the notification reset mechanism. For the first write, RSTNDP = 0x47 while the second write requires RSTNDP = 0xb8.                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 8–15<br>RSTNTN | Reset Notification Number. This 8-bit field specifies the specific IRQ notification state machine to be reset. This field is updated by the second write.<br><br>If RSTNTN < 64, then reset the single IRQ notification machine defined by RSTNTN, else reset all the notifications. |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

## 13.4 Functional Description

In this section, the functional operation of the Semaphores module, specifically the state machines of the SEMA4\_GATE<sub>n</sub> and SEMA4\_CP<sub>n</sub>NTF registers are detailed.

### 13.4.1 SEMA4\_GATEn Operation

Recall each of the SEMA4\_GATEn registers implements a 2-bit, 3-state machine. The state transitions for each gate are shown in the following figure.

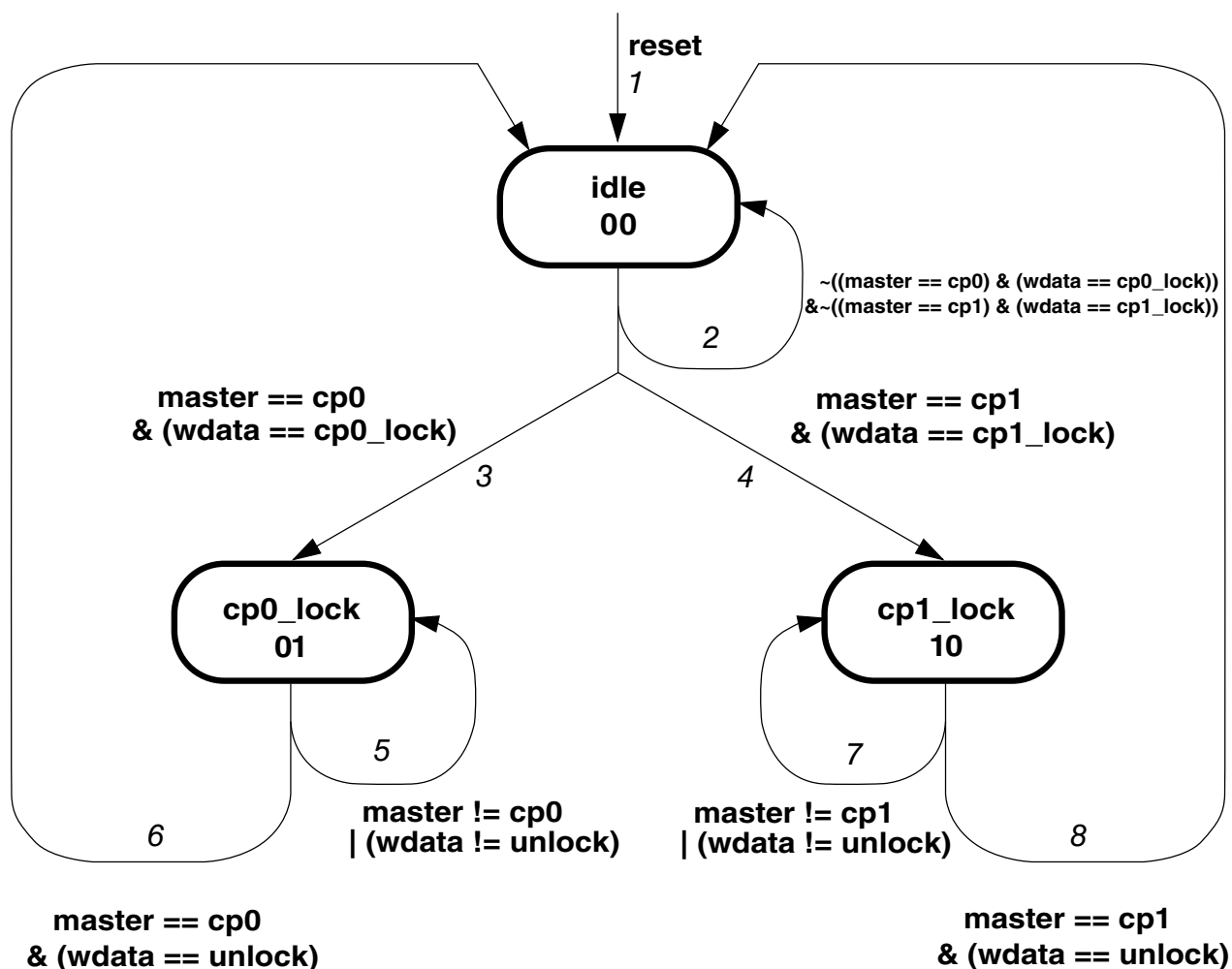


**Figure 13-26. SEMA4\_GATEn State Machine**

The bus master number is used to identify core processor 0 (cp0) or core processor 1 (cp1). The Standard (or Reduced) Product Platform passes the AHB bus master number (`hmaster[2:0]`) through the AIPS (or AIPS-Lite) controller and drives an `aips_master[2:0]` output to the Semaphores module as an IPS sideband signal.

The state transitions for SEMA4\_GATEn are defined in the following table.

**Table 13-26. SEMA4\_GATEn State Transitions**

| Current State | Next State | Transition | Description                                                                                                         |
|---------------|------------|------------|---------------------------------------------------------------------------------------------------------------------|
| —             | idle       | 1          | Any reset, whether a system reset or an individual gate reset, unconditionally forces the gate into the idle state. |

*Table continues on the next page...*



**Table 13-26. SEMA4\_GATEn State Transitions  
(continued)**

| Current State | Next State | Transition | Description                                                                                                               |
|---------------|------------|------------|---------------------------------------------------------------------------------------------------------------------------|
| idle          | idle       | 2          | Unless a write of the appropriate lock value from the corresponding processor occurs, the gate remains in the idle state. |
| idle          | cp0_lock   | 3          | When a write of the "cp0_lock" data value is initiated by processor 0, the gate transitions into the cp0_lock state.      |
| idle          | cp1_lock   | 4          | When a write of the "cp1_lock" value is initiated by processor 1, the gate transitions into the cp1_lock state.           |
| cp0_lock      | cp0_lock   | 5          | Once in this state, the gate remains here if any attempted write is not from cp0 with the unlock data value.              |
| cp0_lock      | idle       | 6          | The gate returns to the idle (unlocked) state once a write from cp0 with the unlock data value occurs.                    |
| cp1_lock      | cp1_lock   | 7          | Once in this state, the gate remains here if any attempted write is not from cp1 with the unlock data value.              |
| cp1_lock      | idle       | 8          | The gate returns to the idle (unlocked) state once a write from cp1 with the unlock data value occurs.                    |

### 13.4.2 SEMA4\_CPnNTF Operation

The failed lock write notification interrupt request is implemented in a 3-bit, 5-state machine which records failed lock attempts and transitions based on gate locking and unlocking. Two specific states are encoded and program-visible as SEMA4\_CP0NTF[GNn] and SEMA4\_CP1NTF[GNn]. See the following figure.

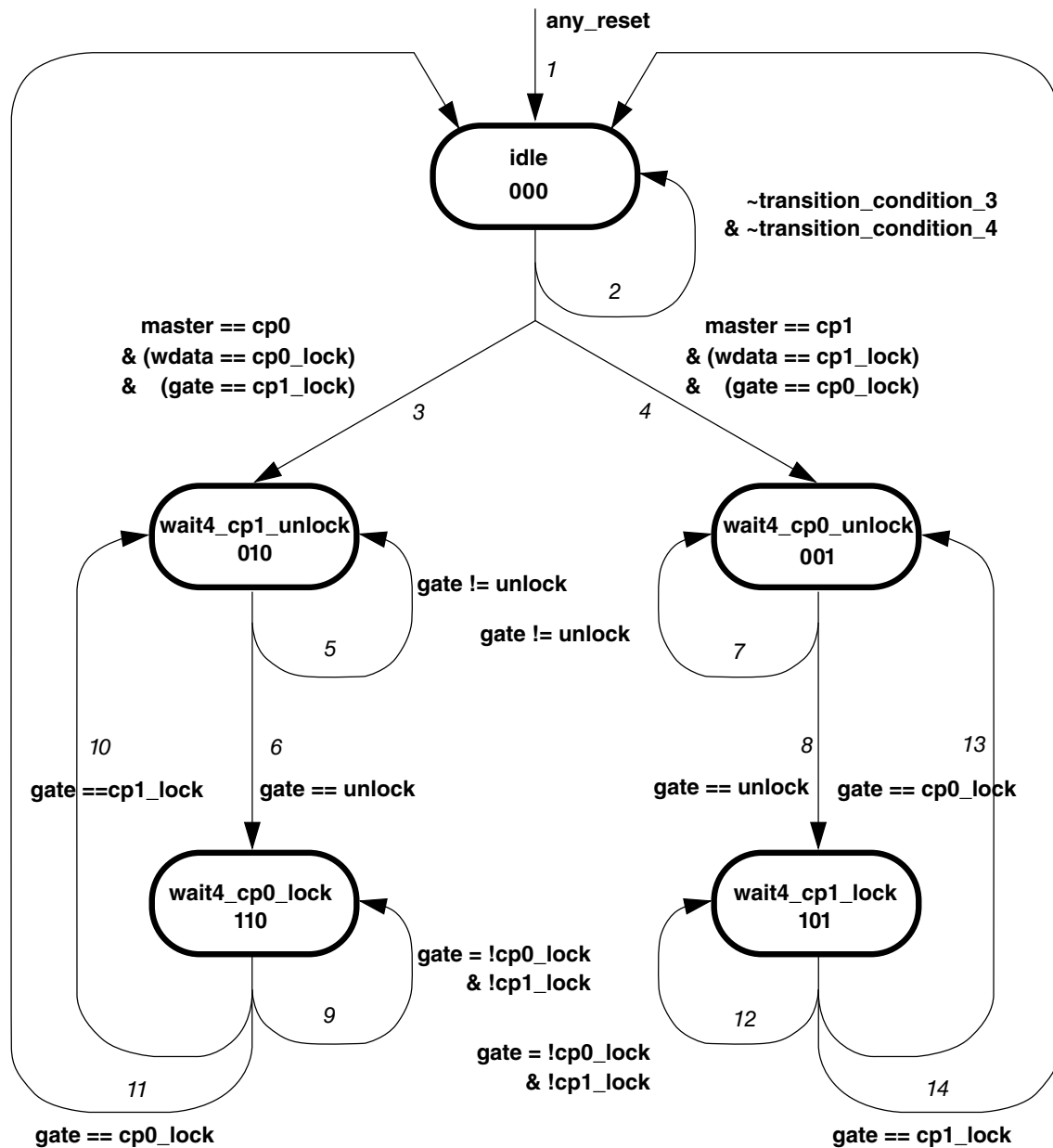


Figure 13-27. IRQ Notification State Machine

The state transitions of the IRQ notification function are defined in the following. Specific states of this machine are program-visible as the SEMA4\_CPnNTF registers. In particular, two states are program-visible:

```

if state = wait4_cp0_lock (0b110) // generate cp0_semaphore_int if properly enabled
    then SEMA4_CP0NTF[GNn] = 1; else SEMA4_CP0NTF[GNn] = 0
if state = wait4_cp1_lock (0b101) // generate cp1_semaphore_int if properly enabled
    then SEMA4_CP1NTF[GNn] = 1; else SEMA4_CP1NTF[GNn] = 0
  
```

**Table 13-27. IRQ Notification State Transitions**

| Current State    | Next State       | Transition | Description                                                                                                                                                                                                              |
|------------------|------------------|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| —                | idle             | 1          | Any reset, including a system reset or an individual notification or secure gate reset, unconditionally forces the machine into the idle state.                                                                          |
| idle             | idle             | 2          | Unless a write of the appropriate lock value from the corresponding processor to an already-locked gate occurs, the machine remains in the idle state.                                                                   |
| idle             | wait4_cp1_unlock | 3          | When a write of the "cp0_lock" data value is initiated by processor 0 but the gate is already locked by cp1, the machine transitions into this state, where it waits for cp1 to unlock the gate.                         |
| idle             | wait4_cp0_unlock | 4          | When a write of the "cp1_lock" data value is initiated by processor 1 but the gate is already locked by cp0, the machine transitions into this state, where it waits for cp0 to unlock the gate.                         |
| wait4_cp1_unlock | wait4_cp1_unlock | 5          | Once in this state, the machine remains here until the gate is unlocked.                                                                                                                                                 |
| wait4_cp1_unlock | wait4_cp0_lock   | 6          | From this state, the machine transitions into the next state, waiting for cp0 to lock the gate, once it has been unlocked.                                                                                               |
| wait4_cp0_unlock | wait4_cp0_unlock | 7          | Once in this state, the machine remains here until the gate is unlocked.                                                                                                                                                 |
| wait4_cp0_unlock | wait4_cp1_lock   | 8          | From this state, the machine transitions into the next state, waiting for cp1 to lock the gate, once it has been unlocked.                                                                                               |
| wait4_cp0_lock   | wait4_cp0_lock   | 9          | In this state, the machine generates the notification interrupt (if properly-enabled) and remains here until the gate is locked by processor 0 or the gate is again locked by processor 1.                               |
| wait4_cp0_lock   | wait4_cp1_unlock | 10         | In this state, the machine generates the notification interrupt (if properly-enabled) and transitions if the gate is again locked by processor 1. With this transition, the notification interrupt request is negated.   |
| wait4_cp0_lock   | idle             | 11         | In this state, the machine generates the notification interrupt (if properly-enabled) and transitions if the gate is finally locked by processor 0. With this transition, the notification interrupt request is negated. |
| wait4_cp1_lock   | wait4_cp1_lock   | 12         | In this state, the machine generates the notification interrupt (if properly-enabled) and remains here until the gate is locked by processor 1 or the gate is again locked by processor 0.                               |
| wait4_cp1_lock   | wait4_cp0_unlock | 13         | In this state, the machine generates the notification interrupt (if properly-enabled) and transitions if the gate is again locked by processor 0. With this transition, the notification interrupt request is negated.   |
| wait4_cp1_lock   | idle             | 14         | In this state, the machine generates the notification interrupt (if properly-enabled) and transitions if the gate is finally locked by processor 1. With this transition, the notification interrupt request is negated. |

The Semaphores module generates two interrupt request output signals, one per processor, combining the SEMA4\_CPnINE and SEMA4\_CPnNTF registers, where the boolean equations are:

```

cp0_semaphore_int
=   sema4_cp0ine[ine0]    &   sema4_cp0ntf[gn0]
  |   sema4_cp0ine[ine1]    &   sema4_cp0ntf[gn1]
  |   sema4_cp0ine[ine2]    &   sema4_cp0ntf[gn2]
  |   ...
  |   sema4_cp0ine[ine15]   &   sema4_cp0ntf[gn15]
cp1_semaphore_int
=   sema4_cp1ine[ine0]    &   sema4_cp1ntf[gn0]
  |   sema4_cp1ine[ine1]    &   sema4_cp1ntf[gn1]
  |   sema4_cp1ine[ine2]    &   sema4_cp1ntf[gn2]
  |   ...
  |   sema4_cp1ine[ine15]   &   sema4_cp1ntf[gn15]

```

## 13.5 Initialization Information

The reset state of the IPS\_Semaphores module allows it to begin operation without the need for any further initialization. All the internal state machines are cleared by any reset event, allowing the module to immediately begin operation.

## 13.6 Application Information

In an operational multi-core system, most interactions involving the Semaphores module involves reads and writes to the SEMA4\_GATEn registers for implementation of the hardware-enforced software gate functions. Typical code segments for gate functions perform the following operations:

- To lock (close) a gate
  - The processor performs a byte write of "logical\_processor\_number + 1" to gate[i]
  - The processor reads back gate[i] and checks for a value of "logical\_processor\_number + 1"

If the compare indicates the expected value, then the gate is locked; proceed with the protected code segment. If the compare does not indicate the expected value, the lock operation failed; repeat the process beginning with byte write to gate[i] in spin-wait loop, or proceed with another execution path and wait for failed lock interrupt notification.

A simple C-language example of a `gateLock` function is shown in the following figure. This function follows the Hennessy/Patterson example described in [Multi-Core Programming 101: Software Gates](#).

```

#define UNLOCK    0
#define CP0_LOCK  1
#define CP1_LOCK  2

void gateLock (n)
int  n;                /* gate number to lock */
{
    int i;
    int current_value;
    int locked_value;

    i = processor_number(); /* obtain logical CPU number */

    if (i == 0)
        locked_value = CP0_LOCK;
    else
        locked_value = CP1_LOCK;

    /* read the current value of the gate and wait until the state == UNLOCK */
    do {
        current_value = gate[n];
    } while (current_value != UNLOCK);

    /* the current value of the gate == UNLOCK. attempt to lock the gate for this
       processor. spin-wait in this loop until gate ownership is obtained */
    do {
        gate[n] = locked_value; /* write gate with processor_number + 1 */
        current_value = gate[n]; /* read gate to verify ownership was obtained */
    } while (current_value != locked_value);
}

```

**Figure 13-28. Sample gateLock Function**

- To unlock (open) a gate
  - After completing the protected code segment, the locking processor performs a byte write of zeroes to gate[i], opening (unlocking) the gate

A few comments on the logical CPU number are appropriate. In this example, a reference to `processor_number()` is used to retrieve this hardware configuration value. Typically, the logical processor numbers are defined by a hardwired input vector to the individual cores. The exact method for accessing the logical processor number varies by architecture. For PowerPC cores, there is a processor ID register (PIR) which is SPR 286 and contains this value. A single instruction can be used to move the contents of the PIR into a general-purpose register: `mf spr rx, 286` where `rx` is the destination GPRn. Other architectures may support a specific instruction to move the contents of the logical processor number into a general-purpose register, e.g., `rdcpn rx` for a "read CPU number" instruction.

If the optional failed lock IRQ notification mechanisms are used, then accesses to the related registers (SEMA4\_CPnINE, SEMA4\_CPnNTF) are required. Note that there is no required negation of the failed lock write notification interrupt as the request is automatically negated by the Semaphores module once the gate has been successfully locked by the "failing" processor.

Finally, in the event a system state requires a software-controlled reset of a gate or IRQ notification register(s), accesses to the secure reset control registers (SEMA4\_RSTGT, SEMA4\_RSTNTF) are required. For these situations, it is recommended that the appropriate IRQ notification enable(s) (SEMA4\_CPnINE) bits be disabled *before* initiating the secure reset 2-write sequence to avoid any race conditions involving spurious notification interrupt requests.

# Chapter 14

## Memory Protection Unit (MPU)

### 14.1 Chip-specific MPU information

This chip has two instances of the MPU module. Some feature details vary between the instances.

The following table summarizes the feature differences. The table does not list feature details that the instances share.

**Table 14-1. MPU instance feature differences**

| Feature                         | MPU_0                  | MPU_1             |
|---------------------------------|------------------------|-------------------|
| Number of slave ports supported | Three numbered 0, 1, 2 | Two numbered 1, 2 |

#### 14.1.1 MPU master and slave assignments

The MPU identifies each XBAR master in terms of the master's logical ID. See the "Logical master ID" assignments in [Table 9-1](#).

Each MPU slave port mediates between an XBAR physical slave port and the module that is assigned to that physical slave port. For each MPU instance, the following tables show the alignment among the MPU slave ports, the XBAR physical slave ports, and the corresponding assigned modules.

**Table 14-2. MPU\_0 slave ports**

| MPU_0 slave port | XBAR physical slave port | Module assigned to XBAR physical slave port |
|------------------|--------------------------|---------------------------------------------|
| 1                | 2                        | PRAMC_0                                     |
| 2                | 6                        | AIPS_1 (Peripheral Bridge A)                |

**Table 14-3. MPU\_1 slave ports**

| MPU_1 slave port | XBAR physical slave port | Module assigned to XBAR physical slave port |
|------------------|--------------------------|---------------------------------------------|
| 1                | 4                        | PRAMC_1                                     |
| 2                | 7                        | AIPS_0 (Peripheral Bridge B)                |

## 14.1.2 MPU register reset values

The following table lists MPU register reset values that are specific to this chip. For the reset values of all other MPU registers, see [Memory map/register definition](#).

**Table 14-4. Chip-specific MPU register reset values**

| Register | MPU_0 and MPU_1 reset value | Notes                                  |
|----------|-----------------------------|----------------------------------------|
| CESR     | 0081_3200h                  | NSP field resets to 0011b <sup>1</sup> |

1. CESR[NSP]'s reset value on MPU\_1 is 3 even though MPU\_1 has only two slave ports.

## 14.2 Introduction

The memory protection unit (MPU) provides hardware access control for all memory references generated in the device.

## 14.3 Overview

The MPU concurrently monitors all system bus transactions and evaluates their appropriateness using pre-programmed region descriptors that define memory spaces and their access rights. Memory references that have sufficient access control rights are allowed to complete, while references that are not mapped to any region descriptor or have insufficient rights are terminated with a protection error response.

### 14.3.1 Block diagram

A simplified block diagram of the MPU module is shown in the following figure.



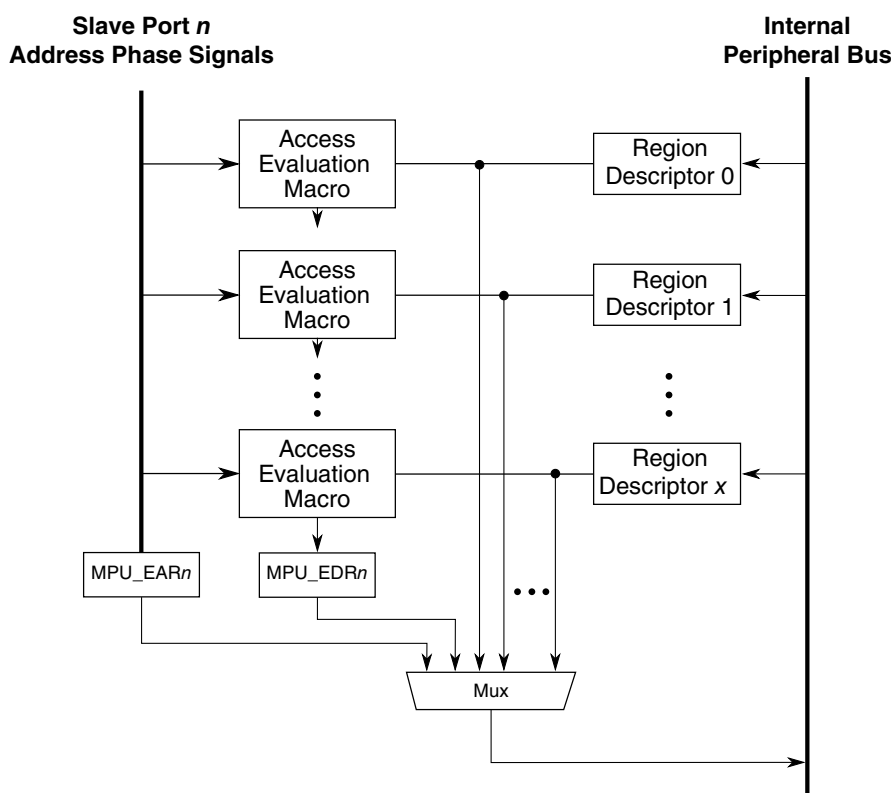


**Figure 14-1. MPU block diagram**

The hardware's two-dimensional connection matrix is clearly visible with the basic access evaluation macro shown as the replicated submodule block. The crossbar switch slave ports are shown on the left, the region descriptor registers in the middle, and the peripheral bus interface on the right side. The evaluation macro contains two magnitude comparators connected to the start and end address registers from each region descriptor as well as the combinational logic blocks to determine the region hit and the access protection error. For details of the access evaluation macro, see [Access evaluation macro](#).

## 14.3.2 Features

The MPU implements a two-dimensional hardware array of memory region descriptors and the crossbar slave ports to continuously monitor the legality of every memory reference generated by each bus master in the system.

The feature set includes:

- 12 program-visible 128-bit region descriptors, accessible by four 32-bit words each
  - Each region descriptor defines a modulo-32 byte space, aligned anywhere in memory

- Region sizes can vary from 32 bytes to 4 Gbytes
- Two access control permissions defined in a single descriptor word
  - Masters 0–3: read, write, and execute attributes for supervisor and user accesses
  - Masters 4–7: read and write attributes
- Hardware-assisted maintenance of the descriptor valid bit minimizes coherency issues
- Alternate programming model view of the access control permissions word
- Priority given to granting permission over denying access for overlapping region descriptors
- Detects access protection errors if a memory reference does not hit in any memory region, or if the reference is illegal in all hit memory regions. If an access error occurs, the reference is terminated with an error response, and the MPU inhibits the bus cycle being sent to the targeted slave device.
- Error registers, per slave port, capture the last faulting address, attributes, and other information
- Global MPU enable/disable control bit

## 14.4 Memory map/register definition

The programming model is partitioned into three groups:

- Control/status registers
- The data structure containing the region descriptors
- The alternate view of the region descriptor access control values

The programming model can be referenced using only 32-bit accesses.

Attempted references of the following types generate an error termination:

- Non-32-bit references
- References to undefined—that is, reserved—addresses
- References with a non-supported access type, such as a write access to a read-only register or a read access of a write-only register

**NOTE**

See the chip configuration details for any chip-specific register information in this module.

**MPU memory map**

| Address offset (hex) | Register name                                   | Width (in bits) | Access | Reset value                 | Section/ page              |
|----------------------|-------------------------------------------------|-----------------|--------|-----------------------------|----------------------------|
| 0                    | Control/Error Status Register (MPU_CESR)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">14.4.1/505</a> |
| 10                   | Error Address Register, slave port n (MPU_EAR0) | 32              | R      | 0000_0000h                  | <a href="#">14.4.2/506</a> |
| 14                   | Error Detail Register, slave port n (MPU_EDR0)  | 32              | R      | 0000_0000h                  | <a href="#">14.4.3/507</a> |
| 18                   | Error Address Register, slave port n (MPU_EAR1) | 32              | R      | 0000_0000h                  | <a href="#">14.4.2/506</a> |
| 1C                   | Error Detail Register, slave port n (MPU_EDR1)  | 32              | R      | 0000_0000h                  | <a href="#">14.4.3/507</a> |
| 20                   | Error Address Register, slave port n (MPU_EAR2) | 32              | R      | 0000_0000h                  | <a href="#">14.4.2/506</a> |
| 24                   | Error Detail Register, slave port n (MPU_EDR2)  | 32              | R      | 0000_0000h                  | <a href="#">14.4.3/507</a> |
| 400                  | Region Descriptor n, Word 0 (MPU_RGD0_WORD0)    | 32              | R/W    | 0000_0000h                  | <a href="#">14.4.4/508</a> |
| 404                  | Region Descriptor n, Word 1 (MPU_RGD0_WORD1)    | 32              | R/W    | <a href="#">See section</a> | <a href="#">14.4.5/508</a> |
| 408                  | Region Descriptor n, Word 2 (MPU_RGD0_WORD2)    | 32              | R/W    | <a href="#">See section</a> | <a href="#">14.4.6/509</a> |
| 40C                  | Region Descriptor n, Word 3 (MPU_RGD0_WORD3)    | 32              | R/W    | 0000_0000h                  | <a href="#">14.4.7/512</a> |
| 410                  | Region Descriptor n, Word 0 (MPU_RGD1_WORD0)    | 32              | R/W    | 0000_0000h                  | <a href="#">14.4.4/508</a> |
| 414                  | Region Descriptor n, Word 1 (MPU_RGD1_WORD1)    | 32              | R/W    | <a href="#">See section</a> | <a href="#">14.4.5/508</a> |
| 418                  | Region Descriptor n, Word 2 (MPU_RGD1_WORD2)    | 32              | R/W    | <a href="#">See section</a> | <a href="#">14.4.6/509</a> |
| 41C                  | Region Descriptor n, Word 3 (MPU_RGD1_WORD3)    | 32              | R/W    | 0000_0000h                  | <a href="#">14.4.7/512</a> |
| 420                  | Region Descriptor n, Word 0 (MPU_RGD2_WORD0)    | 32              | R/W    | 0000_0000h                  | <a href="#">14.4.4/508</a> |
| 424                  | Region Descriptor n, Word 1 (MPU_RGD2_WORD1)    | 32              | R/W    | <a href="#">See section</a> | <a href="#">14.4.5/508</a> |
| 428                  | Region Descriptor n, Word 2 (MPU_RGD2_WORD2)    | 32              | R/W    | <a href="#">See section</a> | <a href="#">14.4.6/509</a> |
| 42C                  | Region Descriptor n, Word 3 (MPU_RGD2_WORD3)    | 32              | R/W    | 0000_0000h                  | <a href="#">14.4.7/512</a> |
| 430                  | Region Descriptor n, Word 0 (MPU_RGD3_WORD0)    | 32              | R/W    | 0000_0000h                  | <a href="#">14.4.4/508</a> |
| 434                  | Region Descriptor n, Word 1 (MPU_RGD3_WORD1)    | 32              | R/W    | <a href="#">See section</a> | <a href="#">14.4.5/508</a> |
| 438                  | Region Descriptor n, Word 2 (MPU_RGD3_WORD2)    | 32              | R/W    | <a href="#">See section</a> | <a href="#">14.4.6/509</a> |
| 43C                  | Region Descriptor n, Word 3 (MPU_RGD3_WORD3)    | 32              | R/W    | 0000_0000h                  | <a href="#">14.4.7/512</a> |
| 440                  | Region Descriptor n, Word 0 (MPU_RGD4_WORD0)    | 32              | R/W    | 0000_0000h                  | <a href="#">14.4.4/508</a> |
| 444                  | Region Descriptor n, Word 1 (MPU_RGD4_WORD1)    | 32              | R/W    | <a href="#">See section</a> | <a href="#">14.4.5/508</a> |
| 448                  | Region Descriptor n, Word 2 (MPU_RGD4_WORD2)    | 32              | R/W    | <a href="#">See section</a> | <a href="#">14.4.6/509</a> |
| 44C                  | Region Descriptor n, Word 3 (MPU_RGD4_WORD3)    | 32              | R/W    | 0000_0000h                  | <a href="#">14.4.7/512</a> |
| 450                  | Region Descriptor n, Word 0 (MPU_RGD5_WORD0)    | 32              | R/W    | 0000_0000h                  | <a href="#">14.4.4/508</a> |
| 454                  | Region Descriptor n, Word 1 (MPU_RGD5_WORD1)    | 32              | R/W    | <a href="#">See section</a> | <a href="#">14.4.5/508</a> |
| 458                  | Region Descriptor n, Word 2 (MPU_RGD5_WORD2)    | 32              | R/W    | <a href="#">See section</a> | <a href="#">14.4.6/509</a> |
| 45C                  | Region Descriptor n, Word 3 (MPU_RGD5_WORD3)    | 32              | R/W    | 0000_0000h                  | <a href="#">14.4.7/512</a> |
| 460                  | Region Descriptor n, Word 0 (MPU_RGD6_WORD0)    | 32              | R/W    | 0000_0000h                  | <a href="#">14.4.4/508</a> |
| 464                  | Region Descriptor n, Word 1 (MPU_RGD6_WORD1)    | 32              | R/W    | <a href="#">See section</a> | <a href="#">14.4.5/508</a> |
| 468                  | Region Descriptor n, Word 2 (MPU_RGD6_WORD2)    | 32              | R/W    | <a href="#">See section</a> | <a href="#">14.4.6/509</a> |
| 46C                  | Region Descriptor n, Word 3 (MPU_RGD6_WORD3)    | 32              | R/W    | 0000_0000h                  | <a href="#">14.4.7/512</a> |

Table continues on the next page...

## MPU memory map (continued)

| Address offset (hex) | Register name                                               | Width (in bits) | Access | Reset value                 | Section/ page              |
|----------------------|-------------------------------------------------------------|-----------------|--------|-----------------------------|----------------------------|
| 470                  | Region Descriptor n, Word 0 (MPU_RGD7_WORD0)                | 32              | R/W    | 0000_0000h                  | <a href="#">14.4.4/508</a> |
| 474                  | Region Descriptor n, Word 1 (MPU_RGD7_WORD1)                | 32              | R/W    | <a href="#">See section</a> | <a href="#">14.4.5/508</a> |
| 478                  | Region Descriptor n, Word 2 (MPU_RGD7_WORD2)                | 32              | R/W    | <a href="#">See section</a> | <a href="#">14.4.6/509</a> |
| 47C                  | Region Descriptor n, Word 3 (MPU_RGD7_WORD3)                | 32              | R/W    | 0000_0000h                  | <a href="#">14.4.7/512</a> |
| 480                  | Region Descriptor n, Word 0 (MPU_RGD8_WORD0)                | 32              | R/W    | 0000_0000h                  | <a href="#">14.4.4/508</a> |
| 484                  | Region Descriptor n, Word 1 (MPU_RGD8_WORD1)                | 32              | R/W    | <a href="#">See section</a> | <a href="#">14.4.5/508</a> |
| 488                  | Region Descriptor n, Word 2 (MPU_RGD8_WORD2)                | 32              | R/W    | <a href="#">See section</a> | <a href="#">14.4.6/509</a> |
| 48C                  | Region Descriptor n, Word 3 (MPU_RGD8_WORD3)                | 32              | R/W    | 0000_0000h                  | <a href="#">14.4.7/512</a> |
| 490                  | Region Descriptor n, Word 0 (MPU_RGD9_WORD0)                | 32              | R/W    | 0000_0000h                  | <a href="#">14.4.4/508</a> |
| 494                  | Region Descriptor n, Word 1 (MPU_RGD9_WORD1)                | 32              | R/W    | <a href="#">See section</a> | <a href="#">14.4.5/508</a> |
| 498                  | Region Descriptor n, Word 2 (MPU_RGD9_WORD2)                | 32              | R/W    | <a href="#">See section</a> | <a href="#">14.4.6/509</a> |
| 49C                  | Region Descriptor n, Word 3 (MPU_RGD9_WORD3)                | 32              | R/W    | 0000_0000h                  | <a href="#">14.4.7/512</a> |
| 4A0                  | Region Descriptor n, Word 0 (MPU_RGD10_WORD0)               | 32              | R/W    | 0000_0000h                  | <a href="#">14.4.4/508</a> |
| 4A4                  | Region Descriptor n, Word 1 (MPU_RGD10_WORD1)               | 32              | R/W    | <a href="#">See section</a> | <a href="#">14.4.5/508</a> |
| 4A8                  | Region Descriptor n, Word 2 (MPU_RGD10_WORD2)               | 32              | R/W    | <a href="#">See section</a> | <a href="#">14.4.6/509</a> |
| 4AC                  | Region Descriptor n, Word 3 (MPU_RGD10_WORD3)               | 32              | R/W    | 0000_0000h                  | <a href="#">14.4.7/512</a> |
| 4B0                  | Region Descriptor n, Word 0 (MPU_RGD11_WORD0)               | 32              | R/W    | 0000_0000h                  | <a href="#">14.4.4/508</a> |
| 4B4                  | Region Descriptor n, Word 1 (MPU_RGD11_WORD1)               | 32              | R/W    | <a href="#">See section</a> | <a href="#">14.4.5/508</a> |
| 4B8                  | Region Descriptor n, Word 2 (MPU_RGD11_WORD2)               | 32              | R/W    | <a href="#">See section</a> | <a href="#">14.4.6/509</a> |
| 4BC                  | Region Descriptor n, Word 3 (MPU_RGD11_WORD3)               | 32              | R/W    | 0000_0000h                  | <a href="#">14.4.7/512</a> |
| 800                  | Region Descriptor Alternate Access Control n (MPU_RGDAAC0)  | 32              | R/W    | <a href="#">See section</a> | <a href="#">14.4.8/513</a> |
| 804                  | Region Descriptor Alternate Access Control n (MPU_RGDAAC1)  | 32              | R/W    | <a href="#">See section</a> | <a href="#">14.4.8/513</a> |
| 808                  | Region Descriptor Alternate Access Control n (MPU_RGDAAC2)  | 32              | R/W    | <a href="#">See section</a> | <a href="#">14.4.8/513</a> |
| 80C                  | Region Descriptor Alternate Access Control n (MPU_RGDAAC3)  | 32              | R/W    | <a href="#">See section</a> | <a href="#">14.4.8/513</a> |
| 810                  | Region Descriptor Alternate Access Control n (MPU_RGDAAC4)  | 32              | R/W    | <a href="#">See section</a> | <a href="#">14.4.8/513</a> |
| 814                  | Region Descriptor Alternate Access Control n (MPU_RGDAAC5)  | 32              | R/W    | <a href="#">See section</a> | <a href="#">14.4.8/513</a> |
| 818                  | Region Descriptor Alternate Access Control n (MPU_RGDAAC6)  | 32              | R/W    | <a href="#">See section</a> | <a href="#">14.4.8/513</a> |
| 81C                  | Region Descriptor Alternate Access Control n (MPU_RGDAAC7)  | 32              | R/W    | <a href="#">See section</a> | <a href="#">14.4.8/513</a> |
| 820                  | Region Descriptor Alternate Access Control n (MPU_RGDAAC8)  | 32              | R/W    | <a href="#">See section</a> | <a href="#">14.4.8/513</a> |
| 824                  | Region Descriptor Alternate Access Control n (MPU_RGDAAC9)  | 32              | R/W    | <a href="#">See section</a> | <a href="#">14.4.8/513</a> |
| 828                  | Region Descriptor Alternate Access Control n (MPU_RGDAAC10) | 32              | R/W    | <a href="#">See section</a> | <a href="#">14.4.8/513</a> |

Table continues on the next page...

## MPU memory map (continued)

| Address offset (hex) | Register name                                               | Width (in bits) | Access | Reset value                 | Section/ page              |
|----------------------|-------------------------------------------------------------|-----------------|--------|-----------------------------|----------------------------|
| 82C                  | Region Descriptor Alternate Access Control n (MPU_RGDAAC11) | 32              | R/W    | <a href="#">See section</a> | <a href="#">14.4.8/513</a> |

## 14.4.1 Control/Error Status Register (MPU\_CESR)

Address: 0h base + 0h offset = 0h

|       |       |   |   |   |   |   |   |   |   |   |    |     |    |    |    |    |
|-------|-------|---|---|---|---|---|---|---|---|---|----|-----|----|----|----|----|
| Bit   | 0     | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11  | 12 | 13 | 14 | 15 |
| R     | SPERR |   |   | 0 |   |   |   | 1 | 0 |   |    | HRL |    |    |    |    |
| W     | w1c   |   |   |   |   |   |   |   |   |   |    |     |    |    |    |    |
| Reset | 0     | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0  | 0   | 0  | 0  | 0  | 1  |

|       |     |    |    |    |      |    |    |    |    |    |    |    |    |    |    |    |     |
|-------|-----|----|----|----|------|----|----|----|----|----|----|----|----|----|----|----|-----|
| Bit   | 16  | 17 | 18 | 19 | 20   | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |     |
| R     | NSP |    |    |    | NRGD |    |    |    | 0  |    |    |    |    |    |    |    | VLD |
| W     |     |    |    |    |      |    |    |    |    |    |    |    |    |    |    |    |     |
| Reset | *   | *  | *  | *  | 0    | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |     |

\* Notes:

- NSP field: See the chip-specific MPU information for the reset value of this field.

## MPU\_CESR field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–2<br>SPERR     | <p>Slave Port n Error</p> <p>Indicates a captured error in EARN and EDNRn. This bit is set when the hardware detects an error and records the faulting address and attributes. It is cleared by writing one to it. If another error is captured at the exact same cycle as the write, the flag remains set. A find-first-one instruction or equivalent can detect the presence of a captured error.</p> <p>The following shows the correspondence between the bit number and slave port number:</p> <ul style="list-style-type: none"> <li>• Bit 0 corresponds to slave port 0.</li> <li>• Bit 1 corresponds to slave port 1.</li> <li>• Bit 2 corresponds to slave port 2.</li> </ul> <p>0 No error has occurred for slave port n.<br/>1 An error has occurred for slave port n.</p> |
| 3–7<br>Reserved  | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 8<br>Reserved    | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 1.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 9–11<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 12–15<br>HRL     | Hardware Revision Level                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |

Table continues on the next page...

**MPU\_CESR field descriptions (continued)**

| Field             | Description                                                                                                                                                                                     |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | Specifies the MPU's hardware and definition revision level. It can be read by software to determine the functional definition of the module.                                                    |
| 16–19<br>NSP      | Number Of Slave Ports<br>Specifies the number of slave ports connected to the MPU.                                                                                                              |
| 20–23<br>NRGD     | Number Of Region Descriptors<br>Indicates the number of region descriptors implemented in the MPU.<br><br>0000 8 region descriptors<br>0001 12 region descriptors<br>0010 16 region descriptors |
| 24–30<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                         |
| 31<br>VLD         | Valid<br>Global enable/disable for the MPU.<br><br>0 MPU is disabled. All accesses from all bus masters are allowed.<br>1 MPU is enabled                                                        |

**14.4.2 Error Address Register, slave port n (MPU\_EAR<sub>n</sub>)**

When the MPU detects an access error on slave port n, the 32-bit reference address is captured in this read-only register and the corresponding bit in CESR[SPERR<sub>n</sub>] is set. Additional information about the faulting access is captured in the corresponding EDR<sub>n</sub> at the same time. This register and the corresponding EDR<sub>n</sub> contain the most recent access error; there are no hardware interlocks with CESR[SPERR<sub>n</sub>], as the error registers are always loaded upon the occurrence of each protection violation.

Address: 0h base + 10h offset + (8d × i), where i=0d to 2d

|       |       |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0     | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | EADDR |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |       |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0     | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**MPU\_EAR<sub>n</sub> field descriptions**

| Field         | Description                                                                                        |
|---------------|----------------------------------------------------------------------------------------------------|
| 0–31<br>EADDR | Error Address<br>Indicates the reference address from slave port n that generated the access error |

### 14.4.3 Error Detail Register, slave port n (MPU\_EDRn)

When the MPU detects an access error on slave port n, 32 bits of error detail are captured in this read-only register and the corresponding bit in CESR[SPERRn] is set. Information on the faulting address is captured in the corresponding EARN register at the same time. This register and the corresponding EARN register contain the most recent access error; there are no hardware interlocks with CESR[SPERRn] as the error registers are always loaded upon the occurrence of each protection violation.

Address: 0h base + 14h offset + (8d × i), where i=0d to 2d

|       |      |    |    |    |    |    |    |    |     |    |    |    |       |    |    |    |
|-------|------|----|----|----|----|----|----|----|-----|----|----|----|-------|----|----|----|
| Bit   | 0    | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8   | 9  | 10 | 11 | 12    | 13 | 14 | 15 |
| R     | EACD |    |    |    |    |    |    |    |     |    |    |    |       |    |    |    |
| W     |      |    |    |    |    |    |    |    |     |    |    |    |       |    |    |    |
| Reset | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0  | 0  | 0  | 0     | 0  | 0  | 0  |
| Bit   | 16   | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24  | 25 | 26 | 27 | 28    | 29 | 30 | 31 |
| R     | EPID |    |    |    |    |    |    |    | EMN |    |    |    | EATTR |    |    |    |
| W     |      |    |    |    |    |    |    |    |     |    |    |    |       |    |    |    |
| Reset | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0  | 0  | 0  | 0     | 0  | 0  | 0  |

**MPU\_EDRn field descriptions**

| Field          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>EACD   | <p>Error Access Control Detail</p> <p>Indicates the region descriptor with the access error.</p> <ul style="list-style-type: none"> <li>• If EDRn contains a captured error and EACD is cleared, an access did not hit in any region descriptor.</li> <li>• If only a single EACD bit is set, the protection error was caused by a single non-overlapping region descriptor.</li> <li>• If two or more EACD bits are set, the protection error was caused by an overlapping set of region descriptors.</li> </ul> |
| 16–23<br>EPID  | <p>Error Process Identification</p> <p>Records the process identifier of the faulting reference. The process identifier is typically driven only by processor cores; for other bus masters, this field is cleared.</p>                                                                                                                                                                                                                                                                                            |
| 24–27<br>EMN   | <p>Error Master Number</p> <p>Indicates the bus master that generated the access error.</p>                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 28–30<br>EATTR | <p>Error Attributes</p> <p>Indicates attribute information about the faulting reference.</p> <p><b>NOTE:</b> All other encodings are reserved.</p> <p>000 User mode, instruction access</p> <p>001 User mode, data access</p> <p>010 Supervisor mode, instruction access</p> <p>011 Supervisor mode, data access</p>                                                                                                                                                                                              |

*Table continues on the next page...*

MPU\_EDR<sub>n</sub> field descriptions (continued)

| Field     | Description                                                                                           |
|-----------|-------------------------------------------------------------------------------------------------------|
| 31<br>ERW | Error Read/Write<br><br>Indicates the access type of the faulting reference.<br><br>0 Read<br>1 Write |

14.4.4 Region Descriptor n, Word 0 (MPU\_RGD<sub>n</sub>\_WORD0)

The first word of the region descriptor defines the 0-modulo-32 byte start address of the memory region. Writes to this register clear the region descriptor's valid bit (RGD<sub>n</sub>\_WORD3[VLD]).

Address: 0h base + 400h offset + (16d × i), where i=0d to 11d

|       |         |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0       | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | SRTADDR |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0  |    |    |    |    |    |
| W     |         |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0       | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

MPU\_RGD<sub>n</sub>\_WORD0 field descriptions

| Field             | Description                                                                                                        |
|-------------------|--------------------------------------------------------------------------------------------------------------------|
| 0–26<br>SRTADDR   | Start Address<br><br>Defines the most significant bits of the 0-modulo-32 byte start address of the memory region. |
| 27–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                            |

14.4.5 Region Descriptor n, Word 1 (MPU\_RGD<sub>n</sub>\_WORD1)

The second word of the region descriptor defines the 31-modulo-32 byte end address of the memory region. Writes to this register clear the region descriptor's valid bit (RGD<sub>n</sub>\_WORD3[VLD]).

Address: 0h base + 404h offset + (16d × i), where i=0d to 11d

|       |         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |
|-------|---------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|----|----|
| Bit   | 0       | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26       | 27 | 28 | 29 | 30 | 31 |
| R     | ENDADDR |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | Reserved |    |    |    |    |    |
| W     |         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |
| Reset | 0*      | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 1*       | 1* | 1* | 1* | 1* | 1* |

\* Notes:

-



**MPU\_RGDn\_WORD1 field descriptions**

| Field             | Description                                                                                                                                                                              |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–26<br>ENDADDR   | End Address<br><br>Defines the most significant bits of the 31-modulo-32 byte end address of the memory region.<br><br><b>NOTE:</b> The MPU does not verify that ENDADDR $\geq$ SRTADDR. |
| 27–31<br>Reserved | This field is reserved.                                                                                                                                                                  |

**14.4.6 Region Descriptor n, Word 2 (MPU\_RGDn\_WORD2)**

The third word of the region descriptor defines the access control rights of the memory region. The access control privileges depend on two broad classifications of bus masters:

- Bus masters 0–3 have a 5-bit field defining separate privilege rights for user and supervisor mode accesses. Each of these bus masters optionally includes a process identification field (if implemented for the master) within the definition.
- Bus masters 4–7 are limited to separate read and write permissions.

For the privilege rights of bus masters 0–3, there are three flags associated with this function:

- Read (r) refers to accessing the referenced memory address using an operand (data) fetch
- Write (w) refers to updating the referenced memory address using a store (data) instruction
- Execute (x) refers to reading the referenced memory address using an instruction fetch

Writes to RGDn\_WORD2 clear the region descriptor's valid bit (RGDn\_WORD3[VLD]). If only updating the access controls, write to RGDAACn instead because stores to these locations do not affect the descriptor's valid bit.

Address: 0h base + 408h offset + (16d  $\times$  i), where i=0d to 11d

| Bit   | 0    | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    | 10 | 11   | 12 | 13 | 14   | 15   |
|-------|------|------|------|------|------|------|------|------|------|------|----|------|----|----|------|------|
| R     | M7RE | M7WE | M6RE | M6WE | M5RE | M5WE | M4RE | M4WE | M3PE | M3SM |    | M3UM |    |    | M2PE | M2SM |
| W     |      |      |      |      |      |      |      |      |      |      |    |      |    |    |      |      |
| Reset | 0*   | 0*   | 0*   | 0*   | 0*   | 0*   | 0*   | 0*   | 0*   | 0*   | 0* | 0*   | 0* | 0* | 0*   | 0*   |

## Memory map/register definition

| Bit   | 16   | 17 | 18 | 19 | 20   | 21 | 22 | 23 | 24   | 25 | 26 | 27 | 28   | 29 | 30 | 31 |
|-------|------|----|----|----|------|----|----|----|------|----|----|----|------|----|----|----|
| R     | M2SM |    |    |    | M2UM |    |    |    | M1PE |    |    |    | M1SM |    |    |    |
| W     |      |    |    |    |      |    |    |    |      |    |    |    |      |    |    |    |
| Reset | 0*   | 0* | 0* | 0* | 0*   | 0* | 0* | 0* | 0*   | 0* | 0* | 0* | 0*   | 0* | 0* | 0* |

\* Notes:

- 

## MPU\_RGDn\_WORD2 field descriptions

| Field     | Description                                                                                                                                                                                |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>M7RE | Bus Master 7 Read Enable<br>0 Bus master 7 reads terminate with an access error and the read is not performed<br>1 Bus master 7 reads allowed                                              |
| 1<br>M7WE | Bus Master 7 Write Enable<br>0 Bus master 7 writes terminate with an access error and the write is not performed<br>1 Bus master 7 writes allowed                                          |
| 2<br>M6RE | Bus Master 6 Read Enable<br>0 Bus master 6 reads terminate with an access error and the read is not performed<br>1 Bus master 6 reads allowed                                              |
| 3<br>M6WE | Bus Master 6 Write Enable<br>0 Bus master 6 writes terminate with an access error and the write is not performed<br>1 Bus master 6 writes allowed                                          |
| 4<br>M5RE | Bus Master 5 Read Enable<br>0 Bus master 5 reads terminate with an access error and the read is not performed<br>1 Bus master 5 reads allowed                                              |
| 5<br>M5WE | Bus Master 5 Write Enable<br>0 Bus master 5 writes terminate with an access error and the write is not performed<br>1 Bus master 5 writes allowed                                          |
| 6<br>M4RE | Bus Master 4 Read Enable<br>0 Bus master 4 reads terminate with an access error and the read is not performed<br>1 Bus master 4 reads allowed                                              |
| 7<br>M4WE | Bus Master 4 Write Enable<br>0 Bus master 4 writes terminate with an access error and the write is not performed<br>1 Bus master 4 writes allowed                                          |
| 8<br>M3PE | Bus Master 3 Process Identifier Enable<br>0 Do not include the process identifier in the evaluation<br>1 Include the process identifier and mask (RGDn_WORD3) in the region hit evaluation |

Table continues on the next page...

**MPU\_RGDn\_WORD2 field descriptions (continued)**

| <b>Field</b>  | <b>Description</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 9–10<br>M3SM  | <p>Bus Master 3 Supervisor Mode Access Control</p> <p>Defines the access controls for bus master 3 in Supervisor mode.</p> <p>00 r/w/x; read, write and execute allowed<br/> 01 r/x; read and execute allowed, but no write<br/> 10 r/w; read and write allowed, but no execute<br/> 11 Same as User mode defined in M3UM</p>                                                                                                                                                                                                                        |
| 11–13<br>M3UM | <p>Bus Master 3 User Mode Access Control</p> <p>Defines the access controls for bus master 3 in User mode. M3UM consists of three independent bits, enabling read (r), write (w), and execute (x) permissions. In M3UM[2:0]: M3UM[2] controls read permissions, M3UM[1] controls write permissions, and M3UM[0] controls execute permissions.</p> <p>0 An attempted access of that mode may be terminated with an access error (if not allowed by another descriptor) and the access not performed.<br/> 1 Allows the given access type to occur</p> |
| 14<br>M2PE    | <p>Bus Master 2 Process Identifier Enable</p> <p>See M3PE description.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 15–16<br>M2SM | <p>Bus Master 2 Supervisor Mode Access Control</p> <p>See M3SM description.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 17–19<br>M2UM | <p>Bus Master 2 User Mode Access control</p> <p>See M3UM description.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 20<br>M1PE    | <p>Bus Master 1 Process Identifier enable</p> <p>See M3PE description.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 21–22<br>M1SM | <p>Bus Master 1 Supervisor Mode Access Control</p> <p>See M3SM description.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 23–25<br>M1UM | <p>Bus Master 1 User Mode Access Control</p> <p>See M3UM description.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26<br>M0PE    | <p>Bus Master 0 Process Identifier enable</p> <p>See M0PE description.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 27–28<br>M0SM | <p>Bus Master 0 Supervisor Mode Access Control</p> <p>See M3SM description.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 29–31<br>M0UM | <p>Bus Master 0 User Mode Access Control</p> <p>See M3UM description.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

14.4.7 Region Descriptor n, Word 3 (MPU\_RGDn\_WORD3)

The fourth word of the region descriptor contains the optional process identifier and mask, plus the region descriptor’s valid bit.

Address: 0h base + 40Ch offset + (16d × i), where i=0d to 11d

|       |     |    |    |    |    |    |    |    |         |    |    |    |    |    |    |     |
|-------|-----|----|----|----|----|----|----|----|---------|----|----|----|----|----|----|-----|
| Bit   | 0   | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8       | 9  | 10 | 11 | 12 | 13 | 14 | 15  |
| R     | PID |    |    |    |    |    |    |    | PIDMASK |    |    |    |    |    |    |     |
| W     |     |    |    |    |    |    |    |    |         |    |    |    |    |    |    |     |
| Reset | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0       | 0  | 0  | 0  | 0  | 0  | 0  | 0   |
| Bit   | 16  | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24      | 25 | 26 | 27 | 28 | 29 | 30 | 31  |
| R     | 0   |    |    |    |    |    |    |    |         |    |    |    |    |    |    | VLD |
| W     |     |    |    |    |    |    |    |    |         |    |    |    |    |    |    |     |
| Reset | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0       | 0  | 0  | 0  | 0  | 0  | 0  | 0   |

MPU\_RGDn\_WORD3 field descriptions

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–7<br>PID        | Process Identifier<br><br>Specifies the process identifier that is included in the region hit determination if RGDn_WORD2[MxPE] is set. PIDMASK can mask individual bits in this field.                                                                                                                                                                                                                                                                   |
| 8–15<br>PIDMASK   | Process Identifier Mask<br><br>Provides a masking capability so that multiple process identifiers can be included as part of the region hit determination. If a bit in PIDMASK is set, then the corresponding PID bit is ignored in the comparison. This field and PID are included in the region hit determination if RGDn_WORD2[MxPE] is set. For more information on the handling of the PID and PIDMASK, see “Access Evaluation - Hit Determination.” |
| 16–30<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                   |
| 31<br>VLD         | Valid<br><br>Signals the region descriptor is valid. Any write to RGDn_WORD0–2 clears this bit.<br><br>0   Region descriptor is invalid<br>1   Region descriptor is valid                                                                                                                                                                                                                                                                                 |

### 14.4.8 Region Descriptor Alternate Access Control n (MPU\_RGDAACn)

Because software may adjust only the access controls within a region descriptor (RGDn\_WORD2) as different tasks execute, an alternate programming view of this 32-bit entity is available. Writing to this register does not affect the descriptor's valid bit.

Address: 0h base + 800h offset + (4d × i), where i=0d to 11d

| Bit   | 0    | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    | 10   | 11   | 12   | 13 | 14 | 15 |
|-------|------|------|------|------|------|------|------|------|------|------|------|------|------|----|----|----|
| R     | M7RE | M7WE | M6RE | M6WE | M5RE | M5WE | M4RE | M4WE | M3PE | M3SM | M3UM | M2PE | M2SM |    |    |    |
| W     |      |      |      |      |      |      |      |      |      |      |      |      |      |    |    |    |
| Reset | 0*   | 0*   | 0*   | 0*   | 0*   | 0*   | 0*   | 0*   | 0*   | 0*   | 0*   | 0*   | 0*   | 0* | 0* | 0* |
| Bit   | 16   | 17   | 18   | 19   | 20   | 21   | 22   | 23   | 24   | 25   | 26   | 27   | 28   | 29 | 30 | 31 |
| R     | M2SM | M2UM | M1PE | M1SM | M1UM | M0PE | M0SM | M0UM |      |      |      |      |      |    |    |    |
| W     |      |      |      |      |      |      |      |      |      |      |      |      |      |    |    |    |
| Reset | 0*   | 0*   | 0*   | 0*   | 0*   | 0*   | 0*   | 0*   | 0*   | 0*   | 0*   | 0*   | 0*   | 0* | 0* | 0* |

\* Notes:

•

#### MPU\_RGDAACn field descriptions

| Field     | Description                                                                                                                                       |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>M7RE | Bus Master 7 Read Enable<br>0 Bus master 7 reads terminate with an access error and the read is not performed<br>1 Bus master 7 reads allowed     |
| 1<br>M7WE | Bus Master 7 Write Enable<br>0 Bus master 7 writes terminate with an access error and the write is not performed<br>1 Bus master 7 writes allowed |
| 2<br>M6RE | Bus Master 6 Read Enable<br>0 Bus master 6 reads terminate with an access error and the read is not performed<br>1 Bus master 6 reads allowed     |
| 3<br>M6WE | Bus Master 6 Write Enable<br>0 Bus master 6 writes terminate with an access error and the write is not performed<br>1 Bus master 6 writes allowed |

Table continues on the next page...

**MPU\_RGDAAC<sub>n</sub> field descriptions (continued)**

| Field         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 4<br>M5RE     | Bus Master 5 Read Enable<br><br>0 Bus master 5 reads terminate with an access error and the read is not performed<br>1 Bus master 5 reads allowed                                                                                                                                                                                                                                                                                                                                                                                           |
| 5<br>M5WE     | Bus Master 5 Write Enable<br><br>0 Bus master 5 writes terminate with an access error and the write is not performed<br>1 Bus master 5 writes allowed                                                                                                                                                                                                                                                                                                                                                                                       |
| 6<br>M4RE     | Bus Master 4 Read Enable<br><br>0 Bus master 4 reads terminate with an access error and the read is not performed<br>1 Bus master 4 reads allowed                                                                                                                                                                                                                                                                                                                                                                                           |
| 7<br>M4WE     | Bus Master 4 Write Enable<br><br>0 Bus master 4 writes terminate with an access error and the write is not performed<br>1 Bus master 4 writes allowed                                                                                                                                                                                                                                                                                                                                                                                       |
| 8<br>M3PE     | Bus Master 3 Process Identifier Enable<br><br>0 Do not include the process identifier in the evaluation<br>1 Include the process identifier and mask (RGDn.RGDAAC) in the region hit evaluation                                                                                                                                                                                                                                                                                                                                             |
| 9–10<br>M3SM  | Bus Master 3 Supervisor Mode Access Control<br><br>Defines the access controls for bus master 3 in Supervisor mode.<br><br>00 r/w/x; read, write and execute allowed<br>01 r/x; read and execute allowed, but no write<br>10 r/w; read and write allowed, but no execute<br>11 Same as User mode defined in M3UM                                                                                                                                                                                                                            |
| 11–13<br>M3UM | Bus Master 3 User Mode Access Control<br><br>Defines the access controls for bus master 3 in user mode. M3UM consists of three independent bits, enabling read (r), write (w), and execute (x) permissions. In M3UM[2:0]: M3UM[2] controls read permissions, M3UM[1] controls write permissions, and M3UM[0] controls execute permissions.<br><br>0 An attempted access of that mode may be terminated with an access error (if not allowed by another descriptor) and the access not performed.<br>1 Allows the given access type to occur |
| 14<br>M2PE    | Bus Master 2 Process Identifier Enable<br><br>See M3PE description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 15–16<br>M2SM | Bus Master 2 Supervisor Mode Access Control<br><br>See M3SM description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 17–19<br>M2UM | Bus Master 2 User Mode Access Control<br><br>See M3UM description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 20<br>M1PE    | Bus Master 1 Process Identifier Enable<br><br>See M3PE description.                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

*Table continues on the next page...*

**MPU\_RGDAAC<sub>n</sub> field descriptions (continued)**

| Field         | Description                                                          |
|---------------|----------------------------------------------------------------------|
| 21–22<br>M1SM | Bus Master 1 Supervisor Mode Access Control<br>See M3SM description. |
| 23–25<br>M1UM | Bus Master 1 User Mode Access Control<br>See M3UM description.       |
| 26<br>M0PE    | Bus Master 0 Process Identifier Enable<br>See M3PE description.      |
| 27–28<br>M0SM | Bus Master 0 Supervisor Mode Access Control<br>See M3SM description. |
| 29–31<br>M0UM | Bus Master 0 User Mode Access Control<br>See M3UM description.       |

## 14.5 Functional description

In this section, the functional operation of the MPU is detailed, including the operation of the access evaluation macro and the handling of error-terminated bus cycles.

### 14.5.1 Access evaluation macro

The basic operation of the MPU is performed in the access evaluation macro, a hardware structure replicated in the two-dimensional connection matrix. As shown in the following figure, the access evaluation macro inputs the crossbar bus address phase signals and the contents of a region descriptor (RGD<sub>n</sub>) and performs two major functions:

- Region hit determination
- Detection of an access protection violation

The following figure shows a functional block diagram.

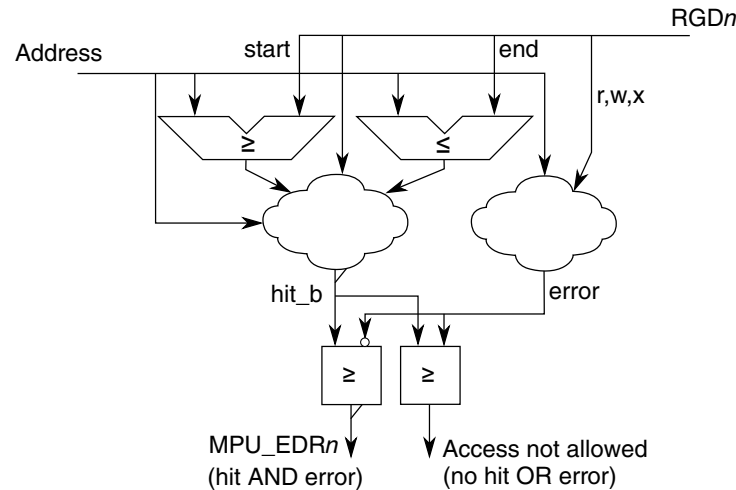


**Figure 14-76. MPU access evaluation macro**

### 14.5.1.1 Hit determination

To determine whether the current reference hits in the given region, two magnitude comparators are used with the region's start and end addresses. The boolean equation for this portion of the hit determination is:

```
region_hit = ((addr[31:5] >= RGDn_Word0[SRTADDR]) & (addr[31:5] <= RGDn_Word1[ENDADDR])) &
RGDn_Word3[VLD]
```

where *addr* is the current reference address, *RGDn\_Word0[SRTADDR]* and *RGDn\_Word1[ENDADDR]* are the start and end addresses, and *RGDn\_Word3[VLD]* is the valid bit.

#### NOTE

The MPU does not verify that *ENDADDR* ≥ *SRTADDR*.

In addition to the comparison of the reference address versus the region descriptor's start and end addresses, the optional process identifier is examined against the region descriptor's PID and PIDMASK fields. A process identifier hit term is formed as follows:

```
pid_hit = ~RGDn_Word2[MxPE] |
((current_pid |
RGDn_Word3[PIDMASK]) == (RGDn_Word3[PID] | RGDn_Word3[PIDMASK]))
```

where the *current\_pid* is the selected process identifier from the current bus master, and *RGDn\_Word3[PID]* and *RGDn\_Word3[PIDMASK]* are the process identifier fields from region descriptor *n*. For bus masters that do not output a process identifier, the MPU forces the *pid\_hit* term to assert.



### 14.5.1.2 Privilege violation determination

While the access evaluation macro is determining region hit, the logic is also evaluating if the current access is allowed by the permissions defined in the region descriptor. Using the master and supervisor/user mode signals, a set of effective permissions is generated from the appropriate fields in the region descriptor. The protection violation logic then evaluates the access against the effective permissions using the specification shown below.

**Table 14-80. Protection violation definition**

| Description            | MxUM |   |   | Protection violation?      |
|------------------------|------|---|---|----------------------------|
|                        | r    | w | x |                            |
| Instruction fetch read | —    | — | 0 | Yes, no execute permission |
|                        | —    | — | 1 | No, access is allowed      |
| Data read              | 0    | — | — | Yes, no read permission    |
|                        | 1    | — | — | No, access is allowed      |
| Data write             | —    | 0 | — | Yes, no write permission   |
|                        | —    | 1 | — | No, access is allowed      |

### 14.5.2 Putting it all together and error terminations

For each slave port monitored, the MPU performs a reduction-AND of all the individual terms from each access evaluation macro. This expression then terminates the bus cycle with an error and reports a protection error for three conditions:

- If the access does not hit in any region descriptor, a protection error is reported.
- If the access hits in a single region descriptor and that region signals a protection violation, a protection error is reported.
- If the access hits in multiple (overlapping) regions and all regions signal protection violations, a protection error is reported.

As shown in the third condition, granting permission is a higher priority than denying access for overlapping regions. This approach is more flexible to system software in region descriptor assignments. For an example of the use of overlapping region descriptors, see [Application information](#).

### 14.5.3 Power management

Disabling the MPU by clearing CESR[VLD] minimizes power dissipation. To minimize the power dissipation of an enabled MPU, invalidate unused region descriptors by clearing the associated RGDn\_Word3[VLD] bits.

## 14.6 Initialization information

At system startup, load the appropriate number of region descriptors, including setting RGDn\_Word3[VLD]. Setting CESR[VLD] enables the module.

If the system requires that all the loaded region descriptors be enabled simultaneously, first ensure that the entire MPU is disabled (CESR[VLD]=0).

### Note

A region descriptor must be set to allow access to the MPU registers if further changes are needed.

## 14.7 Application information

In an operational system, interfacing with the MPU is generally classified into the following activities:

- Creating a new memory region—Load the appropriate region descriptor into an available RGDn, using four sequential 32-bit writes. The hardware assists in the maintenance of the valid bit, so if this approach is followed, there are no coherency issues with the multi-cycle descriptor writes. (Clearing RGDn\_Word3[VLD] deletes/removes an existing memory region.)
- Altering only access privileges—To not affect the valid bit, write to the alternate version of the access control word (RGDAACn), so there are no coherency issues involved with the update. When the write completes, the memory region's access rights switch instantaneously to the new value.
- Changing a region's start and end addresses—Write a minimum of three words to the region descriptor (RGDn\_Word{0,1,3}). Word 0 and 1 redefine the start and end addresses, respectively. Word 3 re-enables the region descriptor valid bit. In most situations, all four words of the region descriptor are rewritten.
- Accessing the MPU—Allocate a region descriptor to restrict MPU access to supervisor mode from a specific master.

- Detecting an access error—The current bus cycle is terminated with an error response and  $EAR_n$  and  $EDR_n$  capture information on the faulting reference. The error-terminated bus cycle typically initiates an error response in the originating bus master. For example, a processor core may respond with a bus error exception, while a data movement bus master may respond with an error interrupt. The processor can retrieve the captured error address and detail information simply by reading  $E\{A,D\}R_n$ .  $CESR[SPERR]$  signals which error registers contain captured fault data.
- Overlapping region descriptors—Applying overlapping regions often reduces the number of descriptors required for a given set of access controls. In the overlapping memory space, the protection rights of the corresponding region descriptors are logically summed together (the boolean OR operator).

The following dual-core system example contains four bus masters:

- The two processors: CP0, CP1
- Two DMA engines: DMA1, a traditional data movement engine transferring data between RAM and peripherals and DMA2, a second engine transferring data to/from the RAM only

Consider the following region descriptor assignments:

**Table 14-81. Overlapping region descriptor example**

| Region description    | RGDn |   | CP0 | CP1 | DMA1 | DMA2 |                  |
|-----------------------|------|---|-----|-----|------|------|------------------|
| CP0 code              | 0    |   | rwX | r-- | —    | —    | Flash            |
| CP1 code              | 1    |   | r-- | rwX | —    | —    |                  |
| CP0 data & stack      | 2    |   | rw- | —   | —    | —    | RAM              |
| CP0 → CP1 shared data | 2    | 3 | r-- | r-- | —    | —    |                  |
| CP1 → CP0 shared data | 4    |   |     |     |      |      |                  |
| CP1 data & stack      | 4    |   | —   | rw- | —    | —    |                  |
| Shared DMA data       | 5    |   | rw- | rw- | rw   | rw   |                  |
| MPU                   | 6    |   | rw- | rw- | —    | —    |                  |
| Peripherals           | 7    |   | rw- | rw- | rw   | —    | Peripheral space |

In this example, there are eight descriptors used to span nine regions in the three main spaces of the system memory map: flash, RAM, and peripheral space. Each region indicates the specific permissions for each of the four bus masters and this definition provides an appropriate set of shared, private and executable memory spaces.

Of particular interest are the two overlapping spaces: region descriptors 2 & 3 and 3 & 4.

The space defined by RGD2 with no overlap is a private data and stack area that provides read/write access to CP0 only. The overlapping space between RGD2 and RGD3 defines a shared data space for passing data from CP0 to CP1 and the access controls are defined by the logical OR of the two region descriptors. Thus, CP0 has (rw- | r--) = (rw-) permissions, while CP1 has (--- | r--) = (r--) permission in this space. Both DMA engines are excluded from this shared processor data region. The overlapping spaces between RGD3 and RGD4 defines another shared data space, this one for passing data from CP1 to CP0. For this overlapping space, CP0 has (r-- | ---) = (r--) permission, while CP1 has (rw- | r--) = (rw-) permission. The non-overlapped space of RGD4 defines a private data and stack area for CP1 only.

The space defined by RGD5 is a shared data region, accessible by all four bus masters. Finally, the slave peripheral space mapped onto the IPS bus is partitioned into two regions:

- One containing the MPU's programming model accessible only to the two processor cores
- The remaining peripheral region accessible to both processors and the traditional DMA1 master

This example shows one possible application of the capabilities of the MPU in a typical system.

# Chapter 15

## Interrupt Controller (INTC)

### 15.1 Chip-specific INTC information

The INTC on this chip supports two core processors. In the detailed INTC description, the cores are designated processor 0 and processor 1.

### 15.2 Introduction

The INTC:

- Provides priority-based preemptive scheduling of interrupt requests
- Schedules interrupt requests (IRQs) from software and internal peripherals to one or both processors (PRCs)
- Provides interrupt prioritization and preemption, interrupt masking, interrupt priority elevation, and protocol support

The INTC has two independent sets of priority arbitration/comparison, request selection, vector encoder and acknowledge logic—one set for each CPU. This arrangement allows each CPU to handle its software-assigned interrupt requests independently of the other CPU's operation, and it provides flexibility for the user to decide which core should handle which interrupt sources in the application. This flexibility comes from a set of configuration bits that allows any interrupt source to generate an interrupt request to Core 0, Core 1, or both cores.

For high-priority interrupt requests in these target applications, it is necessary to minimize the interval between the assertion of the interrupt request from the peripheral and the point at which the processor is performing useful work to service the interrupt request. The INTC supports this goal by providing a unique vector for each interrupt request source. It also provides 16 priorities so that lower priority ISRs do not delay the

execution of higher priority ISRs. Since each individual application will have different priorities for each source of interrupt request, the priority of each interrupt request is configurable.

When multiple tasks share a resource, coherent accesses to that resource need to be supported. The INTC supports the Priority Ceiling Protocol for coherent accesses. By providing a modifiable priority mask, the priority can be raised temporarily so that all tasks which share the resource are unable to preempt each other.

Multiple processors can assert interrupt requests to each other through software-settable interrupt requests. These same software-settable interrupt requests can also be used to separate the work involved in servicing an interrupt request into two parts, a high-priority portion and a low-priority portion. The high-priority portion is initiated by a peripheral interrupt request, but then the ISR can assert a software-settable interrupt request to finish the servicing in a lower priority ISR. Therefore these software-settable interrupt requests can be used instead of having the peripheral ISR schedule a task through the RTOS.

## 15.3 Block diagram

Interrupts implemented by the chip are defined in the *e200z759n3 Core Reference Manual*.

The block diagram for the INTC appears in the following figure.

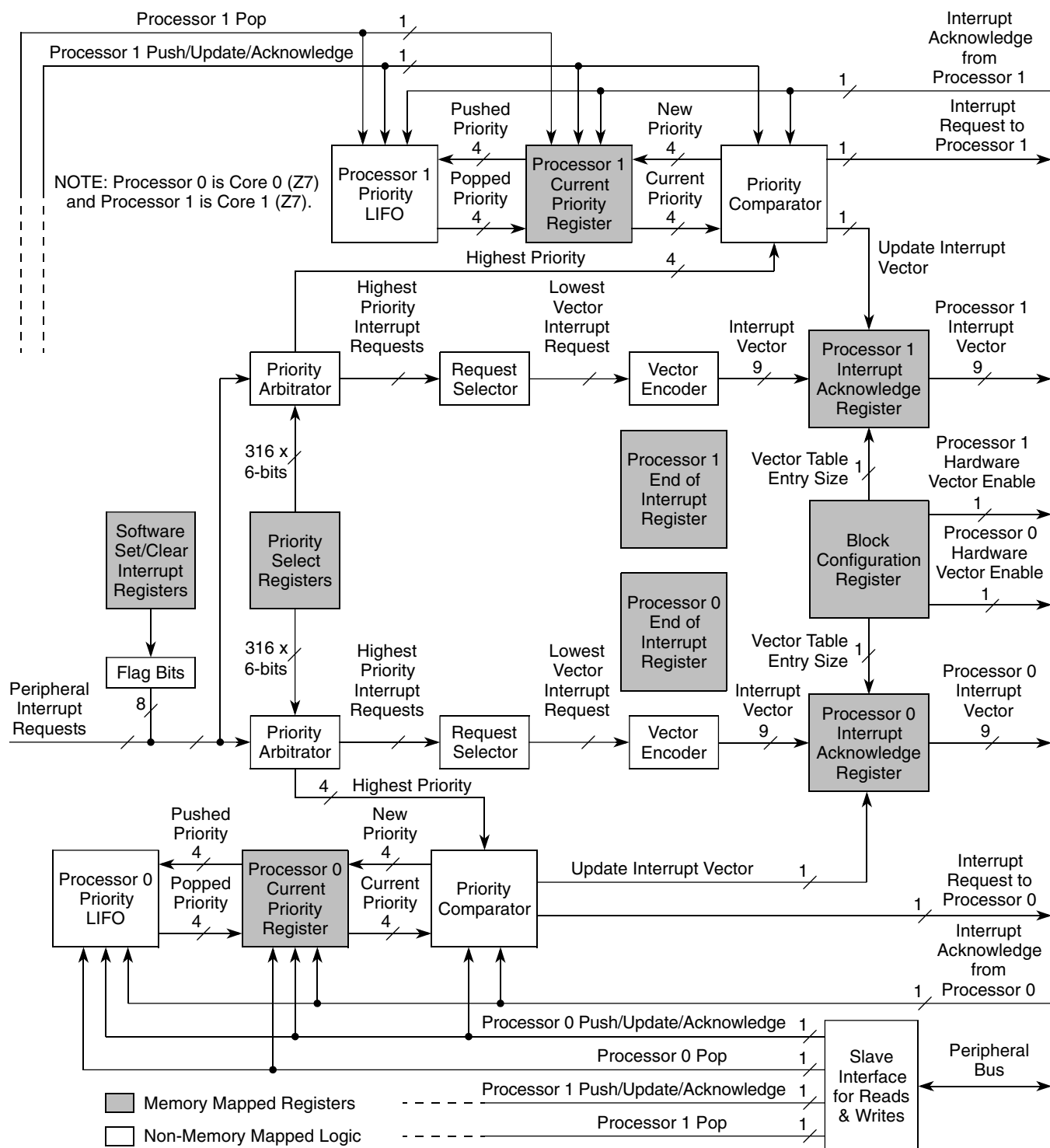


Figure 15-1. INTC block diagram

## 15.4 Features

- Each peripheral interrupt source is software-steerable to interrupt request outputs of processor 0, processor 1, or both
- 8 software-settable interrupt request sources
- 9-bit vector
  - Unique vector for each interrupt request source
  - Hardware connection to processor or read from register
- Each interrupt source can be programmed to one of 16 priorities
- Each interrupt source can be triggered by software
- Preemption
  - Preemptive prioritized interrupt requests to processor
  - ISR with higher priority preempts ISRs or tasks with lower priorities
  - Automatic pushing or popping of preempted priority to or from a LIFO
  - Ability to modify the ISR or task priority; modifying the priority can be used to implement the Priority Ceiling Protocol for accessing shared resources
- Low latency
  - Three INTC clock cycles from receipt of interrupt request from peripheral to interrupt request to processor

## 15.5 Modes of operation

The interrupt controller has two handshaking modes with the processor: software vector mode and hardware vector mode. The state of the hardware vector enable bit, `INTC_MCR[HVEN_PRC $n$ ]`, determines which mode is used for each CPU.

In debug mode the interrupt controller operation is identical to its normal operation of software vector mode or hardware vector mode.



## 15.5.1 Software vector mode

In software vector mode, as shown in the following figure, the CPU branches to a common interrupt exception handler whose location is determined by an address derived from special purpose registers IVPR and IVOR4. The interrupt exception handler reads the INTC\_IACKR\_PCR $n$  to determine the vector of the interrupt request source.

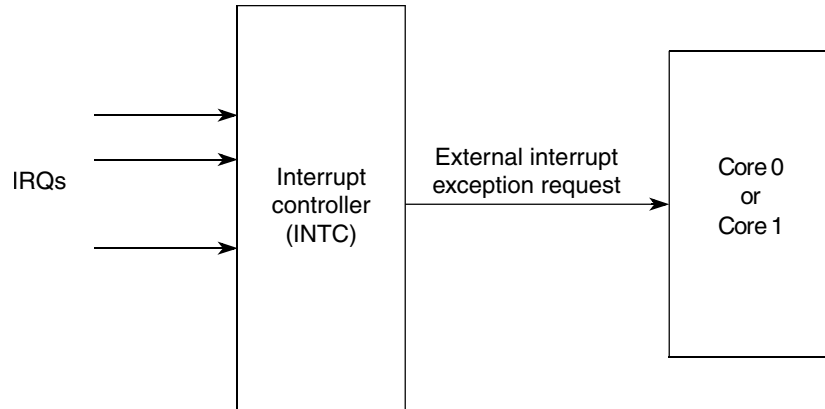
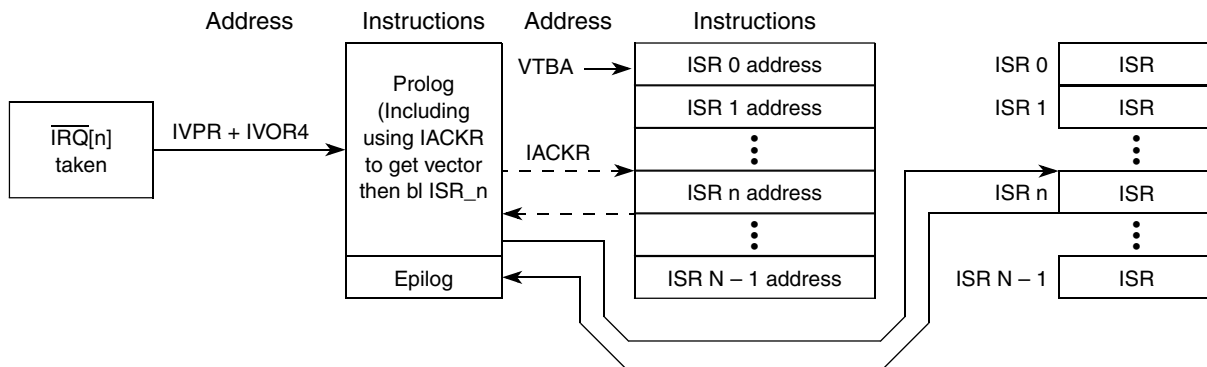


**Figure 15-2. INTC software vector mode**

The following figure shows the typical program flow for software vector mode.



N is the maximum number of usable interrupt vectors and includes eight software-settable IRQ vectors.

**Figure 15-3. Program flow: software vector mode**

The following figure shows how the common interrupt exception handler address is calculated by hardware. The upper half of the interrupt vector prefix register (IVPR) is added to the offset contained in the external input interrupt vector offset register (IVOR4). Because bits IVOR4[28:31] are not part of the offset value, the vector offset must be in a location in memory that is quad-word (16-byte) aligned.

## Modes of operation

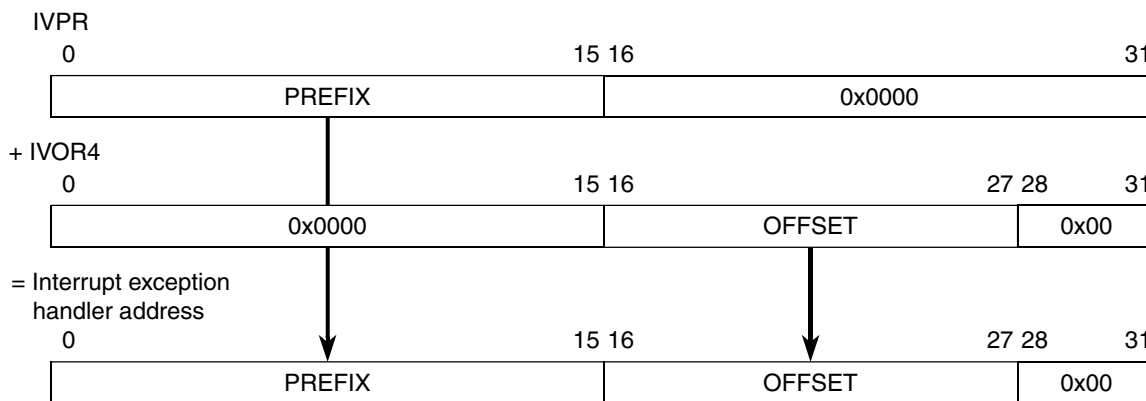


**Figure 15-4. Software vector mode: interrupt exception handler address calculation**

As shown in the preceding figure, the common interrupt exception handler reads the `INTC_IACKR_PRCn` to determine the vector of the interrupt request source. The `INTC_IACKR_PRCn` register contains a 32-bit address for a vector table base address (VTBA) plus an offset to access the interrupt vector (INTVEC). The address is then used to branch to the corresponding routine for that peripheral or software interrupt source.

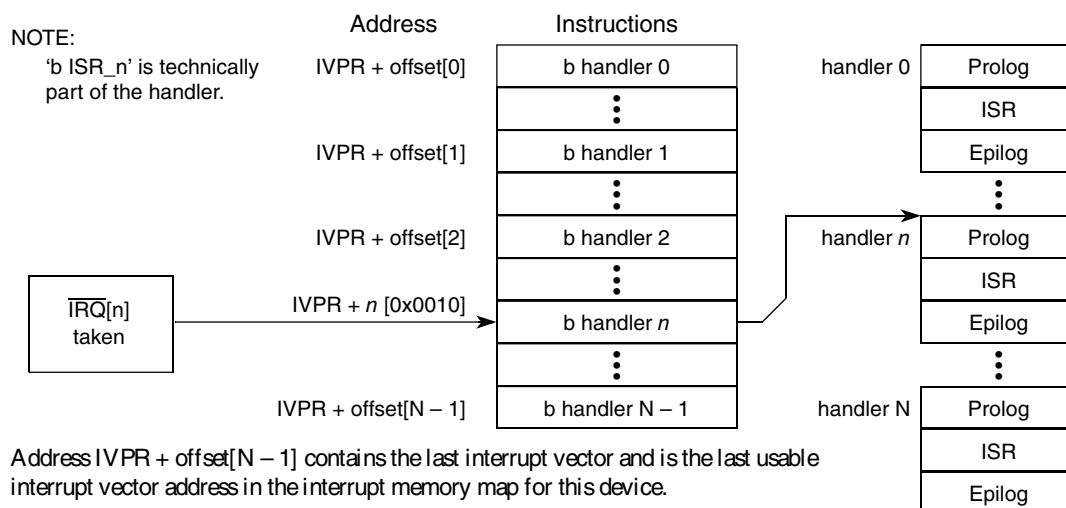
Reading the `INTC_IACKR_PRCn` acknowledges the INTC's interrupt request and negates the interrupt request to the processor. The interrupt request to the processor does not clear if a higher priority interrupt request arrives. Even in this case, INTVEC does not update to the higher priority request until the lower priority interrupt request is acknowledged by reading the `INTC_IACKR_PRCn`. The reading also pushes the PRI value in the INTC current priority register (`INTC_CPR_PRCn`) onto the LIFO and updates PRI in the `INTC_CPR_PRCn` with the priority of the interrupt request. The `INTC_CPR_PRCn` masks any peripheral or software settable interrupt request at the same or lower priority of the current value of the PRI field in `INTC_CPR_PRCn` from generating an interrupt request to the processor.

The interrupt exception handler must write to the end-of-interrupt register (`INTC_EOIR_PRCn`) to complete the operation. Writing to the `INTC_EOIR_PRCn` ends the servicing of the interrupt request. The INTC's LIFO is popped into the `INTC_CPR_PRCn`'s PRI field by writing to the `INTC_EOIR_PRCn`, and the size of a write does not affect the operation of the write. Those values and sizes written to this register neither update the `INTC_EOIR_PRCn` contents nor affect whether the LIFO pops. For possible future compatibility, write four bytes of all zeroes to the `INTC_EOIR_PRCn`. The timing relationship between popping the LIFO and disabling recognition of external input has no restriction. The writes can happen in either order.

However, disabling recognition of the external input before popping the LIFO eases the calculation of the maximum stack depth at the cost of postponing the servicing of the next interrupt request.

## 15.5.2 Hardware vector mode

The following figure shows the typical program flow for hardware vector mode.



N is the maximum number of usable interrupt vectors and includes eight software-settable IRQ vectors.

**Figure 15-5. Program flow: hardware vector mode**

In hardware vector mode, the interrupt exception handler address is specific to the peripheral or software-settable interrupt source rather than being common to all of them. No IVOR is used. The interrupt exception handler address is calculated by hardware as shown in the following figure. The upper half of the interrupt vector prefix register (IVPR) is added to an offset, which corresponds to the peripheral or software interrupt source which caused the interrupt request. The offset matches the value in the Interrupt Vector field, INTC\_IACKR\_PRCn[INTVEC]. Each interrupt exception handler address is aligned on a quad word (16-byte) boundary. IVOR4 is not used in this mode, and software does not need to read INTC\_IACKR\_PRCn to get the interrupt vector number.

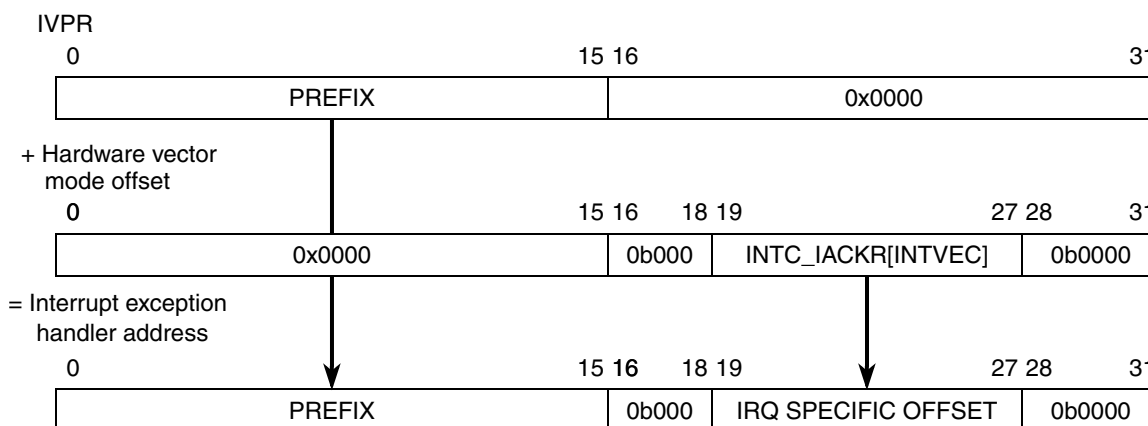


**Figure 15-6. Hardware vector mode: interrupt exception handler address calculation**

The processor negates INTC's interrupt request when automatically acknowledging the interrupt request. However, the interrupt request to the processor do not negate if a higher priority interrupt request arrives. Even in this case, the interrupt vector number does not update to the higher priority request until the lower priority request is acknowledged by the processor.

The assertion of the interrupt acknowledge signal pushes the PRI value in the INTC\_CPR\_PRC $n$  onto the LIFO and updates PRI in the INTC\_CPR\_PRC $n$  with the new priority.

## 15.6 External signal description

The INTC has no direct external MCU signals. However, there are external pins that can be configured in the SIU as external interrupt request input pins. When configured in this function, an interrupt on the pin sets an external interrupt flag. These flags can cause one of five peripheral interrupt requests to the interrupt controller.

## 15.7 Memory map and register definition

The actual availability and number of registers are chip-specific. For this information, see the chapter that describes how modules are configured and connected.

### NOTE

For unimplemented locations in the memory map, read accesses return zeros and attempted write accesses are ignored.

The INTC register memory map is mirrored at 1 KB intervals. Referring to the following equation, the result of accessing any location when I is 1 to 15 is the same as the result of accessing a location when I is 0:

Offset + (400h x I), where I = 0 to 15

For example: PSR0 is accessed at 040h, 440h, 840h, and so on.

With the exception of the INTC\_SSCIR $n$  and INTC\_PSR $n$  registers, all of the registers are 32-bit. Any combination of accessing the four bytes of a register with a single access is supported, provided that the access does not cross a register boundary. These supported accesses include types and sizes of 8 bits, aligned 16 bits, misaligned 16 bits to the middle two bytes, and aligned 32 bits.

Although INTC\_SSCIR $n$  and INTC\_PSR $n$  are 8 bits wide, they can be accessed with a single 16-bit or 32-bit access, provided that the access does not cross a 32-bit boundary.

In software vector mode, the effects of a read of the INTC Interrupt Acknowledge register are the same regardless of the size of the read. In either software or hardware vector mode, the size of a write to the INTC End-of-Interrupt register does not affect the operation of the write.

### INTC memory map

| Address offset (hex) | Register name                                                         | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|-----------------------------------------------------------------------|-----------------|--------|-------------|----------------------------|
| 0                    | INTC Module Configuration Register (INTC_MCR)                         | 32              | R/W    | 0000_0000h  | <a href="#">15.7.1/543</a> |
| 8                    | INTC Current Priority Register for Processor n (INTC_CPR_PRC0)        | 32              | R/W    | 0000_000Fh  | <a href="#">15.7.2/544</a> |
| C                    | INTC Current Priority Register for Processor n (INTC_CPR_PRC1)        | 32              | R/W    | 0000_000Fh  | <a href="#">15.7.2/544</a> |
| 10                   | INTC Interrupt Acknowledge Register for Processor n (INTC_IACKR_PRC0) | 32              | R/W    | 0000_0000h  | <a href="#">15.7.3/545</a> |
| 14                   | INTC Interrupt Acknowledge Register for Processor n (INTC_IACKR_PRC1) | 32              | R/W    | 0000_0000h  | <a href="#">15.7.3/545</a> |
| 18                   | INTC End Of Interrupt Register for Processor n (INTC_EOIR_PRC0)       | 32              | W      | 0000_0000h  | <a href="#">15.7.4/546</a> |
| 1C                   | INTC End Of Interrupt Register for Processor n (INTC_EOIR_PRC1)       | 32              | W      | 0000_0000h  | <a href="#">15.7.4/546</a> |
| 20                   | INTC Software Set/Clear Interrupt Register (INTC_SSCIR0)              | 8               | R/W    | 00h         | <a href="#">15.7.5/547</a> |
| 21                   | INTC Software Set/Clear Interrupt Register (INTC_SSCIR1)              | 8               | R/W    | 00h         | <a href="#">15.7.5/547</a> |
| 22                   | INTC Software Set/Clear Interrupt Register (INTC_SSCIR2)              | 8               | R/W    | 00h         | <a href="#">15.7.5/547</a> |
| 23                   | INTC Software Set/Clear Interrupt Register (INTC_SSCIR3)              | 8               | R/W    | 00h         | <a href="#">15.7.5/547</a> |
| 24                   | INTC Software Set/Clear Interrupt Register (INTC_SSCIR4)              | 8               | R/W    | 00h         | <a href="#">15.7.5/547</a> |
| 25                   | INTC Software Set/Clear Interrupt Register (INTC_SSCIR5)              | 8               | R/W    | 00h         | <a href="#">15.7.5/547</a> |
| 26                   | INTC Software Set/Clear Interrupt Register (INTC_SSCIR6)              | 8               | R/W    | 00h         | <a href="#">15.7.5/547</a> |
| 27                   | INTC Software Set/Clear Interrupt Register (INTC_SSCIR7)              | 8               | R/W    | 00h         | <a href="#">15.7.5/547</a> |
| 40                   | INTC Priority Select Register (INTC_PSR0)                             | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 41                   | INTC Priority Select Register (INTC_PSR1)                             | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 42                   | INTC Priority Select Register (INTC_PSR2)                             | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 43                   | INTC Priority Select Register (INTC_PSR3)                             | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 44                   | INTC Priority Select Register (INTC_PSR4)                             | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 45                   | INTC Priority Select Register (INTC_PSR5)                             | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 46                   | INTC Priority Select Register (INTC_PSR6)                             | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 47                   | INTC Priority Select Register (INTC_PSR7)                             | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 48                   | INTC Priority Select Register (INTC_PSR8)                             | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 49                   | INTC Priority Select Register (INTC_PSR9)                             | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 4A                   | INTC Priority Select Register (INTC_PSR10)                            | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 4B                   | INTC Priority Select Register (INTC_PSR11)                            | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |

Table continues on the next page...

## INTC memory map (continued)

| Address offset (hex) | Register name                              | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|--------------------------------------------|-----------------|--------|-------------|----------------------------|
| 4C                   | INTC Priority Select Register (INTC_PSR12) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 4D                   | INTC Priority Select Register (INTC_PSR13) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 4E                   | INTC Priority Select Register (INTC_PSR14) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 4F                   | INTC Priority Select Register (INTC_PSR15) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 50                   | INTC Priority Select Register (INTC_PSR16) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 51                   | INTC Priority Select Register (INTC_PSR17) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 52                   | INTC Priority Select Register (INTC_PSR18) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 53                   | INTC Priority Select Register (INTC_PSR19) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 54                   | INTC Priority Select Register (INTC_PSR20) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 55                   | INTC Priority Select Register (INTC_PSR21) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 56                   | INTC Priority Select Register (INTC_PSR22) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 57                   | INTC Priority Select Register (INTC_PSR23) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 58                   | INTC Priority Select Register (INTC_PSR24) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 59                   | INTC Priority Select Register (INTC_PSR25) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 5A                   | INTC Priority Select Register (INTC_PSR26) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 5B                   | INTC Priority Select Register (INTC_PSR27) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 5C                   | INTC Priority Select Register (INTC_PSR28) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 5D                   | INTC Priority Select Register (INTC_PSR29) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 5E                   | INTC Priority Select Register (INTC_PSR30) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 5F                   | INTC Priority Select Register (INTC_PSR31) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 60                   | INTC Priority Select Register (INTC_PSR32) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 61                   | INTC Priority Select Register (INTC_PSR33) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 62                   | INTC Priority Select Register (INTC_PSR34) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 63                   | INTC Priority Select Register (INTC_PSR35) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 64                   | INTC Priority Select Register (INTC_PSR36) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 65                   | INTC Priority Select Register (INTC_PSR37) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 66                   | INTC Priority Select Register (INTC_PSR38) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 67                   | INTC Priority Select Register (INTC_PSR39) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 68                   | INTC Priority Select Register (INTC_PSR40) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 69                   | INTC Priority Select Register (INTC_PSR41) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 6A                   | INTC Priority Select Register (INTC_PSR42) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 6B                   | INTC Priority Select Register (INTC_PSR43) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 6C                   | INTC Priority Select Register (INTC_PSR44) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 6D                   | INTC Priority Select Register (INTC_PSR45) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 6E                   | INTC Priority Select Register (INTC_PSR46) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 6F                   | INTC Priority Select Register (INTC_PSR47) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 70                   | INTC Priority Select Register (INTC_PSR48) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 71                   | INTC Priority Select Register (INTC_PSR49) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 72                   | INTC Priority Select Register (INTC_PSR50) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |

Table continues on the next page...

## INTC memory map (continued)

| Address offset (hex) | Register name                              | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|--------------------------------------------|-----------------|--------|-------------|----------------------------|
| 73                   | INTC Priority Select Register (INTC_PSR51) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 74                   | INTC Priority Select Register (INTC_PSR52) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 75                   | INTC Priority Select Register (INTC_PSR53) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 76                   | INTC Priority Select Register (INTC_PSR54) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 77                   | INTC Priority Select Register (INTC_PSR55) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 78                   | INTC Priority Select Register (INTC_PSR56) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 79                   | INTC Priority Select Register (INTC_PSR57) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 7A                   | INTC Priority Select Register (INTC_PSR58) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 7B                   | INTC Priority Select Register (INTC_PSR59) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 7C                   | INTC Priority Select Register (INTC_PSR60) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 7D                   | INTC Priority Select Register (INTC_PSR61) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 7E                   | INTC Priority Select Register (INTC_PSR62) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 7F                   | INTC Priority Select Register (INTC_PSR63) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 80                   | INTC Priority Select Register (INTC_PSR64) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 81                   | INTC Priority Select Register (INTC_PSR65) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 82                   | INTC Priority Select Register (INTC_PSR66) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 83                   | INTC Priority Select Register (INTC_PSR67) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 84                   | INTC Priority Select Register (INTC_PSR68) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 85                   | INTC Priority Select Register (INTC_PSR69) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 86                   | INTC Priority Select Register (INTC_PSR70) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 87                   | INTC Priority Select Register (INTC_PSR71) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 88                   | INTC Priority Select Register (INTC_PSR72) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 89                   | INTC Priority Select Register (INTC_PSR73) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 8A                   | INTC Priority Select Register (INTC_PSR74) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 8B                   | INTC Priority Select Register (INTC_PSR75) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 8C                   | INTC Priority Select Register (INTC_PSR76) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 8D                   | INTC Priority Select Register (INTC_PSR77) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 8E                   | INTC Priority Select Register (INTC_PSR78) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 8F                   | INTC Priority Select Register (INTC_PSR79) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 90                   | INTC Priority Select Register (INTC_PSR80) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 91                   | INTC Priority Select Register (INTC_PSR81) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 92                   | INTC Priority Select Register (INTC_PSR82) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 93                   | INTC Priority Select Register (INTC_PSR83) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 94                   | INTC Priority Select Register (INTC_PSR84) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 95                   | INTC Priority Select Register (INTC_PSR85) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 96                   | INTC Priority Select Register (INTC_PSR86) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 97                   | INTC Priority Select Register (INTC_PSR87) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 98                   | INTC Priority Select Register (INTC_PSR88) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 99                   | INTC Priority Select Register (INTC_PSR89) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |

Table continues on the next page...

## INTC memory map (continued)

| Address offset (hex) | Register name                               | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|---------------------------------------------|-----------------|--------|-------------|----------------------------|
| 9A                   | INTC Priority Select Register (INTC_PSR90)  | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 9B                   | INTC Priority Select Register (INTC_PSR91)  | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 9C                   | INTC Priority Select Register (INTC_PSR92)  | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 9D                   | INTC Priority Select Register (INTC_PSR93)  | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 9E                   | INTC Priority Select Register (INTC_PSR94)  | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 9F                   | INTC Priority Select Register (INTC_PSR95)  | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| A0                   | INTC Priority Select Register (INTC_PSR96)  | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| A1                   | INTC Priority Select Register (INTC_PSR97)  | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| A2                   | INTC Priority Select Register (INTC_PSR98)  | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| A3                   | INTC Priority Select Register (INTC_PSR99)  | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| A4                   | INTC Priority Select Register (INTC_PSR100) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| A5                   | INTC Priority Select Register (INTC_PSR101) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| A6                   | INTC Priority Select Register (INTC_PSR102) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| A7                   | INTC Priority Select Register (INTC_PSR103) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| A8                   | INTC Priority Select Register (INTC_PSR104) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| A9                   | INTC Priority Select Register (INTC_PSR105) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| AA                   | INTC Priority Select Register (INTC_PSR106) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| AB                   | INTC Priority Select Register (INTC_PSR107) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| AC                   | INTC Priority Select Register (INTC_PSR108) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| AD                   | INTC Priority Select Register (INTC_PSR109) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| AE                   | INTC Priority Select Register (INTC_PSR110) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| AF                   | INTC Priority Select Register (INTC_PSR111) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| B0                   | INTC Priority Select Register (INTC_PSR112) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| B1                   | INTC Priority Select Register (INTC_PSR113) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| B2                   | INTC Priority Select Register (INTC_PSR114) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| B3                   | INTC Priority Select Register (INTC_PSR115) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| B4                   | INTC Priority Select Register (INTC_PSR116) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| B5                   | INTC Priority Select Register (INTC_PSR117) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| B6                   | INTC Priority Select Register (INTC_PSR118) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| B7                   | INTC Priority Select Register (INTC_PSR119) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| B8                   | INTC Priority Select Register (INTC_PSR120) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| B9                   | INTC Priority Select Register (INTC_PSR121) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| BA                   | INTC Priority Select Register (INTC_PSR122) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| BB                   | INTC Priority Select Register (INTC_PSR123) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| BC                   | INTC Priority Select Register (INTC_PSR124) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| BD                   | INTC Priority Select Register (INTC_PSR125) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| BE                   | INTC Priority Select Register (INTC_PSR126) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| BF                   | INTC Priority Select Register (INTC_PSR127) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| C0                   | INTC Priority Select Register (INTC_PSR128) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |

Table continues on the next page...



## INTC memory map (continued)

| Address offset (hex) | Register name                               | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|---------------------------------------------|-----------------|--------|-------------|----------------------------|
| C1                   | INTC Priority Select Register (INTC_PSR129) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| C2                   | INTC Priority Select Register (INTC_PSR130) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| C3                   | INTC Priority Select Register (INTC_PSR131) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| C4                   | INTC Priority Select Register (INTC_PSR132) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| C5                   | INTC Priority Select Register (INTC_PSR133) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| C6                   | INTC Priority Select Register (INTC_PSR134) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| C7                   | INTC Priority Select Register (INTC_PSR135) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| C8                   | INTC Priority Select Register (INTC_PSR136) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| C9                   | INTC Priority Select Register (INTC_PSR137) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| CA                   | INTC Priority Select Register (INTC_PSR138) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| CB                   | INTC Priority Select Register (INTC_PSR139) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| CC                   | INTC Priority Select Register (INTC_PSR140) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| CD                   | INTC Priority Select Register (INTC_PSR141) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| CE                   | INTC Priority Select Register (INTC_PSR142) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| CF                   | INTC Priority Select Register (INTC_PSR143) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| D0                   | INTC Priority Select Register (INTC_PSR144) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| D1                   | INTC Priority Select Register (INTC_PSR145) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| D2                   | INTC Priority Select Register (INTC_PSR146) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| D3                   | INTC Priority Select Register (INTC_PSR147) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| D4                   | INTC Priority Select Register (INTC_PSR148) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| D5                   | INTC Priority Select Register (INTC_PSR149) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| D6                   | INTC Priority Select Register (INTC_PSR150) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| D7                   | INTC Priority Select Register (INTC_PSR151) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| D8                   | INTC Priority Select Register (INTC_PSR152) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| D9                   | INTC Priority Select Register (INTC_PSR153) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| DA                   | INTC Priority Select Register (INTC_PSR154) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| DB                   | INTC Priority Select Register (INTC_PSR155) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| DC                   | INTC Priority Select Register (INTC_PSR156) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| DD                   | INTC Priority Select Register (INTC_PSR157) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| DE                   | INTC Priority Select Register (INTC_PSR158) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| DF                   | INTC Priority Select Register (INTC_PSR159) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| E0                   | INTC Priority Select Register (INTC_PSR160) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| E1                   | INTC Priority Select Register (INTC_PSR161) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| E2                   | INTC Priority Select Register (INTC_PSR162) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| E3                   | INTC Priority Select Register (INTC_PSR163) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| E4                   | INTC Priority Select Register (INTC_PSR164) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| E5                   | INTC Priority Select Register (INTC_PSR165) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| E6                   | INTC Priority Select Register (INTC_PSR166) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| E7                   | INTC Priority Select Register (INTC_PSR167) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |

Table continues on the next page...

## INTC memory map (continued)

| Address offset (hex) | Register name                               | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|---------------------------------------------|-----------------|--------|-------------|----------------------------|
| E8                   | INTC Priority Select Register (INTC_PSR168) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| E9                   | INTC Priority Select Register (INTC_PSR169) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| EA                   | INTC Priority Select Register (INTC_PSR170) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| EB                   | INTC Priority Select Register (INTC_PSR171) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| EC                   | INTC Priority Select Register (INTC_PSR172) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| ED                   | INTC Priority Select Register (INTC_PSR173) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| EE                   | INTC Priority Select Register (INTC_PSR174) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| EF                   | INTC Priority Select Register (INTC_PSR175) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| F0                   | INTC Priority Select Register (INTC_PSR176) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| F1                   | INTC Priority Select Register (INTC_PSR177) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| F2                   | INTC Priority Select Register (INTC_PSR178) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| F3                   | INTC Priority Select Register (INTC_PSR179) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| F4                   | INTC Priority Select Register (INTC_PSR180) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| F5                   | INTC Priority Select Register (INTC_PSR181) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| F6                   | INTC Priority Select Register (INTC_PSR182) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| F7                   | INTC Priority Select Register (INTC_PSR183) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| F8                   | INTC Priority Select Register (INTC_PSR184) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| F9                   | INTC Priority Select Register (INTC_PSR185) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| FA                   | INTC Priority Select Register (INTC_PSR186) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| FB                   | INTC Priority Select Register (INTC_PSR187) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| FC                   | INTC Priority Select Register (INTC_PSR188) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| FD                   | INTC Priority Select Register (INTC_PSR189) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| FE                   | INTC Priority Select Register (INTC_PSR190) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| FF                   | INTC Priority Select Register (INTC_PSR191) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 100                  | INTC Priority Select Register (INTC_PSR192) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 101                  | INTC Priority Select Register (INTC_PSR193) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 102                  | INTC Priority Select Register (INTC_PSR194) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 103                  | INTC Priority Select Register (INTC_PSR195) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 104                  | INTC Priority Select Register (INTC_PSR196) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 105                  | INTC Priority Select Register (INTC_PSR197) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 106                  | INTC Priority Select Register (INTC_PSR198) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 107                  | INTC Priority Select Register (INTC_PSR199) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 108                  | INTC Priority Select Register (INTC_PSR200) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 109                  | INTC Priority Select Register (INTC_PSR201) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 10A                  | INTC Priority Select Register (INTC_PSR202) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 10B                  | INTC Priority Select Register (INTC_PSR203) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 10C                  | INTC Priority Select Register (INTC_PSR204) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 10D                  | INTC Priority Select Register (INTC_PSR205) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 10E                  | INTC Priority Select Register (INTC_PSR206) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |

Table continues on the next page...

## INTC memory map (continued)

| Address offset (hex) | Register name                               | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|---------------------------------------------|-----------------|--------|-------------|----------------------------|
| 10F                  | INTC Priority Select Register (INTC_PSR207) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 110                  | INTC Priority Select Register (INTC_PSR208) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 111                  | INTC Priority Select Register (INTC_PSR209) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 112                  | INTC Priority Select Register (INTC_PSR210) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 113                  | INTC Priority Select Register (INTC_PSR211) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 114                  | INTC Priority Select Register (INTC_PSR212) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 115                  | INTC Priority Select Register (INTC_PSR213) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 116                  | INTC Priority Select Register (INTC_PSR214) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 117                  | INTC Priority Select Register (INTC_PSR215) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 118                  | INTC Priority Select Register (INTC_PSR216) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 119                  | INTC Priority Select Register (INTC_PSR217) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 11A                  | INTC Priority Select Register (INTC_PSR218) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 11B                  | INTC Priority Select Register (INTC_PSR219) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 11C                  | INTC Priority Select Register (INTC_PSR220) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 11D                  | INTC Priority Select Register (INTC_PSR221) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 11E                  | INTC Priority Select Register (INTC_PSR222) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 11F                  | INTC Priority Select Register (INTC_PSR223) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 120                  | INTC Priority Select Register (INTC_PSR224) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 121                  | INTC Priority Select Register (INTC_PSR225) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 122                  | INTC Priority Select Register (INTC_PSR226) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 123                  | INTC Priority Select Register (INTC_PSR227) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 124                  | INTC Priority Select Register (INTC_PSR228) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 125                  | INTC Priority Select Register (INTC_PSR229) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 126                  | INTC Priority Select Register (INTC_PSR230) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 127                  | INTC Priority Select Register (INTC_PSR231) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 128                  | INTC Priority Select Register (INTC_PSR232) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 129                  | INTC Priority Select Register (INTC_PSR233) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 12A                  | INTC Priority Select Register (INTC_PSR234) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 12B                  | INTC Priority Select Register (INTC_PSR235) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 12C                  | INTC Priority Select Register (INTC_PSR236) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 12D                  | INTC Priority Select Register (INTC_PSR237) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 12E                  | INTC Priority Select Register (INTC_PSR238) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 12F                  | INTC Priority Select Register (INTC_PSR239) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 130                  | INTC Priority Select Register (INTC_PSR240) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 131                  | INTC Priority Select Register (INTC_PSR241) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 132                  | INTC Priority Select Register (INTC_PSR242) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 133                  | INTC Priority Select Register (INTC_PSR243) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 134                  | INTC Priority Select Register (INTC_PSR244) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 135                  | INTC Priority Select Register (INTC_PSR245) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |

Table continues on the next page...

## INTC memory map (continued)

| Address offset (hex) | Register name                               | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|---------------------------------------------|-----------------|--------|-------------|----------------------------|
| 136                  | INTC Priority Select Register (INTC_PSR246) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 137                  | INTC Priority Select Register (INTC_PSR247) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 138                  | INTC Priority Select Register (INTC_PSR248) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 139                  | INTC Priority Select Register (INTC_PSR249) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 13A                  | INTC Priority Select Register (INTC_PSR250) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 13B                  | INTC Priority Select Register (INTC_PSR251) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 13C                  | INTC Priority Select Register (INTC_PSR252) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 13D                  | INTC Priority Select Register (INTC_PSR253) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 13E                  | INTC Priority Select Register (INTC_PSR254) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 13F                  | INTC Priority Select Register (INTC_PSR255) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 140                  | INTC Priority Select Register (INTC_PSR256) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 141                  | INTC Priority Select Register (INTC_PSR257) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 142                  | INTC Priority Select Register (INTC_PSR258) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 143                  | INTC Priority Select Register (INTC_PSR259) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 144                  | INTC Priority Select Register (INTC_PSR260) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 145                  | INTC Priority Select Register (INTC_PSR261) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 146                  | INTC Priority Select Register (INTC_PSR262) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 147                  | INTC Priority Select Register (INTC_PSR263) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 148                  | INTC Priority Select Register (INTC_PSR264) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 149                  | INTC Priority Select Register (INTC_PSR265) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 14A                  | INTC Priority Select Register (INTC_PSR266) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 14B                  | INTC Priority Select Register (INTC_PSR267) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 14C                  | INTC Priority Select Register (INTC_PSR268) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 14D                  | INTC Priority Select Register (INTC_PSR269) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 14E                  | INTC Priority Select Register (INTC_PSR270) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 14F                  | INTC Priority Select Register (INTC_PSR271) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 150                  | INTC Priority Select Register (INTC_PSR272) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 151                  | INTC Priority Select Register (INTC_PSR273) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 152                  | INTC Priority Select Register (INTC_PSR274) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 153                  | INTC Priority Select Register (INTC_PSR275) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 154                  | INTC Priority Select Register (INTC_PSR276) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 155                  | INTC Priority Select Register (INTC_PSR277) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 156                  | INTC Priority Select Register (INTC_PSR278) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 157                  | INTC Priority Select Register (INTC_PSR279) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 158                  | INTC Priority Select Register (INTC_PSR280) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 159                  | INTC Priority Select Register (INTC_PSR281) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 15A                  | INTC Priority Select Register (INTC_PSR282) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 15B                  | INTC Priority Select Register (INTC_PSR283) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 15C                  | INTC Priority Select Register (INTC_PSR284) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |

Table continues on the next page...

## INTC memory map (continued)

| Address offset (hex) | Register name                               | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|---------------------------------------------|-----------------|--------|-------------|----------------------------|
| 15D                  | INTC Priority Select Register (INTC_PSR285) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 15E                  | INTC Priority Select Register (INTC_PSR286) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 15F                  | INTC Priority Select Register (INTC_PSR287) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 160                  | INTC Priority Select Register (INTC_PSR288) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 161                  | INTC Priority Select Register (INTC_PSR289) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 162                  | INTC Priority Select Register (INTC_PSR290) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 163                  | INTC Priority Select Register (INTC_PSR291) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 164                  | INTC Priority Select Register (INTC_PSR292) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 165                  | INTC Priority Select Register (INTC_PSR293) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 166                  | INTC Priority Select Register (INTC_PSR294) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 167                  | INTC Priority Select Register (INTC_PSR295) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 168                  | INTC Priority Select Register (INTC_PSR296) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 169                  | INTC Priority Select Register (INTC_PSR297) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 16A                  | INTC Priority Select Register (INTC_PSR298) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 16B                  | INTC Priority Select Register (INTC_PSR299) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 16C                  | INTC Priority Select Register (INTC_PSR300) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 16D                  | INTC Priority Select Register (INTC_PSR301) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 16E                  | INTC Priority Select Register (INTC_PSR302) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 16F                  | INTC Priority Select Register (INTC_PSR303) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 170                  | INTC Priority Select Register (INTC_PSR304) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 171                  | INTC Priority Select Register (INTC_PSR305) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 172                  | INTC Priority Select Register (INTC_PSR306) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 173                  | INTC Priority Select Register (INTC_PSR307) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 174                  | INTC Priority Select Register (INTC_PSR308) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 175                  | INTC Priority Select Register (INTC_PSR309) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 176                  | INTC Priority Select Register (INTC_PSR310) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 177                  | INTC Priority Select Register (INTC_PSR311) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 178                  | INTC Priority Select Register (INTC_PSR312) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 179                  | INTC Priority Select Register (INTC_PSR313) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 17A                  | INTC Priority Select Register (INTC_PSR314) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 17B                  | INTC Priority Select Register (INTC_PSR315) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 17C                  | INTC Priority Select Register (INTC_PSR316) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 17D                  | INTC Priority Select Register (INTC_PSR317) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 17E                  | INTC Priority Select Register (INTC_PSR318) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 17F                  | INTC Priority Select Register (INTC_PSR319) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 180                  | INTC Priority Select Register (INTC_PSR320) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 181                  | INTC Priority Select Register (INTC_PSR321) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 182                  | INTC Priority Select Register (INTC_PSR322) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 183                  | INTC Priority Select Register (INTC_PSR323) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |

Table continues on the next page...



## INTC memory map (continued)

| Address offset (hex) | Register name                               | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|---------------------------------------------|-----------------|--------|-------------|----------------------------|
| 184                  | INTC Priority Select Register (INTC_PSR324) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 185                  | INTC Priority Select Register (INTC_PSR325) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 186                  | INTC Priority Select Register (INTC_PSR326) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 187                  | INTC Priority Select Register (INTC_PSR327) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 188                  | INTC Priority Select Register (INTC_PSR328) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 189                  | INTC Priority Select Register (INTC_PSR329) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 18A                  | INTC Priority Select Register (INTC_PSR330) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 18B                  | INTC Priority Select Register (INTC_PSR331) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 18C                  | INTC Priority Select Register (INTC_PSR332) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 18D                  | INTC Priority Select Register (INTC_PSR333) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 18E                  | INTC Priority Select Register (INTC_PSR334) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 18F                  | INTC Priority Select Register (INTC_PSR335) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 190                  | INTC Priority Select Register (INTC_PSR336) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 191                  | INTC Priority Select Register (INTC_PSR337) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 192                  | INTC Priority Select Register (INTC_PSR338) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 193                  | INTC Priority Select Register (INTC_PSR339) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 194                  | INTC Priority Select Register (INTC_PSR340) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 195                  | INTC Priority Select Register (INTC_PSR341) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 196                  | INTC Priority Select Register (INTC_PSR342) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 197                  | INTC Priority Select Register (INTC_PSR343) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 198                  | INTC Priority Select Register (INTC_PSR344) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 199                  | INTC Priority Select Register (INTC_PSR345) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 19A                  | INTC Priority Select Register (INTC_PSR346) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 19B                  | INTC Priority Select Register (INTC_PSR347) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 19C                  | INTC Priority Select Register (INTC_PSR348) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 19D                  | INTC Priority Select Register (INTC_PSR349) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 19E                  | INTC Priority Select Register (INTC_PSR350) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 19F                  | INTC Priority Select Register (INTC_PSR351) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1A0                  | INTC Priority Select Register (INTC_PSR352) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1A1                  | INTC Priority Select Register (INTC_PSR353) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1A2                  | INTC Priority Select Register (INTC_PSR354) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1A3                  | INTC Priority Select Register (INTC_PSR355) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1A4                  | INTC Priority Select Register (INTC_PSR356) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1A5                  | INTC Priority Select Register (INTC_PSR357) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1A6                  | INTC Priority Select Register (INTC_PSR358) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1A7                  | INTC Priority Select Register (INTC_PSR359) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1A8                  | INTC Priority Select Register (INTC_PSR360) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1A9                  | INTC Priority Select Register (INTC_PSR361) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1AA                  | INTC Priority Select Register (INTC_PSR362) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |

Table continues on the next page...

## INTC memory map (continued)

| Address offset (hex) | Register name                               | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|---------------------------------------------|-----------------|--------|-------------|----------------------------|
| 1AB                  | INTC Priority Select Register (INTC_PSR363) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1AC                  | INTC Priority Select Register (INTC_PSR364) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1AD                  | INTC Priority Select Register (INTC_PSR365) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1AE                  | INTC Priority Select Register (INTC_PSR366) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1AF                  | INTC Priority Select Register (INTC_PSR367) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1B0                  | INTC Priority Select Register (INTC_PSR368) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1B1                  | INTC Priority Select Register (INTC_PSR369) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1B2                  | INTC Priority Select Register (INTC_PSR370) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1B3                  | INTC Priority Select Register (INTC_PSR371) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1B4                  | INTC Priority Select Register (INTC_PSR372) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1B5                  | INTC Priority Select Register (INTC_PSR373) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1B6                  | INTC Priority Select Register (INTC_PSR374) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1B7                  | INTC Priority Select Register (INTC_PSR375) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1B8                  | INTC Priority Select Register (INTC_PSR376) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1B9                  | INTC Priority Select Register (INTC_PSR377) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1BA                  | INTC Priority Select Register (INTC_PSR378) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1BB                  | INTC Priority Select Register (INTC_PSR379) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1BC                  | INTC Priority Select Register (INTC_PSR380) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1BD                  | INTC Priority Select Register (INTC_PSR381) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1BE                  | INTC Priority Select Register (INTC_PSR382) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1BF                  | INTC Priority Select Register (INTC_PSR383) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1C0                  | INTC Priority Select Register (INTC_PSR384) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1C1                  | INTC Priority Select Register (INTC_PSR385) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1C2                  | INTC Priority Select Register (INTC_PSR386) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1C3                  | INTC Priority Select Register (INTC_PSR387) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1C4                  | INTC Priority Select Register (INTC_PSR388) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1C5                  | INTC Priority Select Register (INTC_PSR389) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1C6                  | INTC Priority Select Register (INTC_PSR390) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1C7                  | INTC Priority Select Register (INTC_PSR391) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1C8                  | INTC Priority Select Register (INTC_PSR392) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1C9                  | INTC Priority Select Register (INTC_PSR393) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1CA                  | INTC Priority Select Register (INTC_PSR394) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1CB                  | INTC Priority Select Register (INTC_PSR395) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1CC                  | INTC Priority Select Register (INTC_PSR396) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1CD                  | INTC Priority Select Register (INTC_PSR397) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1CE                  | INTC Priority Select Register (INTC_PSR398) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1CF                  | INTC Priority Select Register (INTC_PSR399) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1D0                  | INTC Priority Select Register (INTC_PSR400) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1D1                  | INTC Priority Select Register (INTC_PSR401) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |

Table continues on the next page...

## INTC memory map (continued)

| Address offset (hex) | Register name                               | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|---------------------------------------------|-----------------|--------|-------------|----------------------------|
| 1D2                  | INTC Priority Select Register (INTC_PSR402) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1D3                  | INTC Priority Select Register (INTC_PSR403) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1D4                  | INTC Priority Select Register (INTC_PSR404) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1D5                  | INTC Priority Select Register (INTC_PSR405) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1D6                  | INTC Priority Select Register (INTC_PSR406) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1D7                  | INTC Priority Select Register (INTC_PSR407) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1D8                  | INTC Priority Select Register (INTC_PSR408) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1D9                  | INTC Priority Select Register (INTC_PSR409) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1DA                  | INTC Priority Select Register (INTC_PSR410) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1DB                  | INTC Priority Select Register (INTC_PSR411) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1DC                  | INTC Priority Select Register (INTC_PSR412) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1DD                  | INTC Priority Select Register (INTC_PSR413) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1DE                  | INTC Priority Select Register (INTC_PSR414) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1DF                  | INTC Priority Select Register (INTC_PSR415) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1E0                  | INTC Priority Select Register (INTC_PSR416) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1E1                  | INTC Priority Select Register (INTC_PSR417) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1E2                  | INTC Priority Select Register (INTC_PSR418) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1E3                  | INTC Priority Select Register (INTC_PSR419) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1E4                  | INTC Priority Select Register (INTC_PSR420) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1E5                  | INTC Priority Select Register (INTC_PSR421) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1E6                  | INTC Priority Select Register (INTC_PSR422) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1E7                  | INTC Priority Select Register (INTC_PSR423) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1E8                  | INTC Priority Select Register (INTC_PSR424) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1E9                  | INTC Priority Select Register (INTC_PSR425) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1EA                  | INTC Priority Select Register (INTC_PSR426) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1EB                  | INTC Priority Select Register (INTC_PSR427) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1EC                  | INTC Priority Select Register (INTC_PSR428) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1ED                  | INTC Priority Select Register (INTC_PSR429) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1EE                  | INTC Priority Select Register (INTC_PSR430) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1EF                  | INTC Priority Select Register (INTC_PSR431) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1F0                  | INTC Priority Select Register (INTC_PSR432) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1F1                  | INTC Priority Select Register (INTC_PSR433) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1F2                  | INTC Priority Select Register (INTC_PSR434) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1F3                  | INTC Priority Select Register (INTC_PSR435) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1F4                  | INTC Priority Select Register (INTC_PSR436) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1F5                  | INTC Priority Select Register (INTC_PSR437) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1F6                  | INTC Priority Select Register (INTC_PSR438) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1F7                  | INTC Priority Select Register (INTC_PSR439) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1F8                  | INTC Priority Select Register (INTC_PSR440) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |

Table continues on the next page...



## INTC memory map (continued)

| Address offset (hex) | Register name                               | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|---------------------------------------------|-----------------|--------|-------------|----------------------------|
| 1F9                  | INTC Priority Select Register (INTC_PSR441) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1FA                  | INTC Priority Select Register (INTC_PSR442) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1FB                  | INTC Priority Select Register (INTC_PSR443) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1FC                  | INTC Priority Select Register (INTC_PSR444) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1FD                  | INTC Priority Select Register (INTC_PSR445) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1FE                  | INTC Priority Select Register (INTC_PSR446) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 1FF                  | INTC Priority Select Register (INTC_PSR447) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 200                  | INTC Priority Select Register (INTC_PSR448) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 201                  | INTC Priority Select Register (INTC_PSR449) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 202                  | INTC Priority Select Register (INTC_PSR450) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 203                  | INTC Priority Select Register (INTC_PSR451) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 204                  | INTC Priority Select Register (INTC_PSR452) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 205                  | INTC Priority Select Register (INTC_PSR453) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 206                  | INTC Priority Select Register (INTC_PSR454) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 207                  | INTC Priority Select Register (INTC_PSR455) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 208                  | INTC Priority Select Register (INTC_PSR456) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 209                  | INTC Priority Select Register (INTC_PSR457) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 20A                  | INTC Priority Select Register (INTC_PSR458) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 20B                  | INTC Priority Select Register (INTC_PSR459) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 20C                  | INTC Priority Select Register (INTC_PSR460) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 20D                  | INTC Priority Select Register (INTC_PSR461) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 20E                  | INTC Priority Select Register (INTC_PSR462) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 20F                  | INTC Priority Select Register (INTC_PSR463) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 210                  | INTC Priority Select Register (INTC_PSR464) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 211                  | INTC Priority Select Register (INTC_PSR465) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 212                  | INTC Priority Select Register (INTC_PSR466) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 213                  | INTC Priority Select Register (INTC_PSR467) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 214                  | INTC Priority Select Register (INTC_PSR468) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 215                  | INTC Priority Select Register (INTC_PSR469) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 216                  | INTC Priority Select Register (INTC_PSR470) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 217                  | INTC Priority Select Register (INTC_PSR471) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 218                  | INTC Priority Select Register (INTC_PSR472) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 219                  | INTC Priority Select Register (INTC_PSR473) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 21A                  | INTC Priority Select Register (INTC_PSR474) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 21B                  | INTC Priority Select Register (INTC_PSR475) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 21C                  | INTC Priority Select Register (INTC_PSR476) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 21D                  | INTC Priority Select Register (INTC_PSR477) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 21E                  | INTC Priority Select Register (INTC_PSR478) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 21F                  | INTC Priority Select Register (INTC_PSR479) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |

Table continues on the next page...

## INTC memory map (continued)

| Address offset (hex) | Register name                               | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|---------------------------------------------|-----------------|--------|-------------|----------------------------|
| 220                  | INTC Priority Select Register (INTC_PSR480) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 221                  | INTC Priority Select Register (INTC_PSR481) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 222                  | INTC Priority Select Register (INTC_PSR482) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 223                  | INTC Priority Select Register (INTC_PSR483) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 224                  | INTC Priority Select Register (INTC_PSR484) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 225                  | INTC Priority Select Register (INTC_PSR485) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 226                  | INTC Priority Select Register (INTC_PSR486) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 227                  | INTC Priority Select Register (INTC_PSR487) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 228                  | INTC Priority Select Register (INTC_PSR488) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 229                  | INTC Priority Select Register (INTC_PSR489) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 22A                  | INTC Priority Select Register (INTC_PSR490) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 22B                  | INTC Priority Select Register (INTC_PSR491) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 22C                  | INTC Priority Select Register (INTC_PSR492) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 22D                  | INTC Priority Select Register (INTC_PSR493) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 22E                  | INTC Priority Select Register (INTC_PSR494) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 22F                  | INTC Priority Select Register (INTC_PSR495) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 230                  | INTC Priority Select Register (INTC_PSR496) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 231                  | INTC Priority Select Register (INTC_PSR497) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 232                  | INTC Priority Select Register (INTC_PSR498) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 233                  | INTC Priority Select Register (INTC_PSR499) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 234                  | INTC Priority Select Register (INTC_PSR500) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 235                  | INTC Priority Select Register (INTC_PSR501) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 236                  | INTC Priority Select Register (INTC_PSR502) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 237                  | INTC Priority Select Register (INTC_PSR503) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 238                  | INTC Priority Select Register (INTC_PSR504) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 239                  | INTC Priority Select Register (INTC_PSR505) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 23A                  | INTC Priority Select Register (INTC_PSR506) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 23B                  | INTC Priority Select Register (INTC_PSR507) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 23C                  | INTC Priority Select Register (INTC_PSR508) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 23D                  | INTC Priority Select Register (INTC_PSR509) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 23E                  | INTC Priority Select Register (INTC_PSR510) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |
| 23F                  | INTC Priority Select Register (INTC_PSR511) | 8               | R/W    | 00h         | <a href="#">15.7.6/548</a> |

## 15.7.1 INTC Module Configuration Register (INTC\_MCR)

The Block Configuration Register is used to configure options of the INTC.

Address: 0h base + 0h offset = 0h

|       |    |           |    |    |    |    |    |           |    |    |           |    |    |    |    |           |
|-------|----|-----------|----|----|----|----|----|-----------|----|----|-----------|----|----|----|----|-----------|
| Bit   | 0  | 1         | 2  | 3  | 4  | 5  | 6  | 7         | 8  | 9  | 10        | 11 | 12 | 13 | 14 | 15        |
| R     | 0  |           |    |    |    |    |    |           |    |    |           |    |    |    |    |           |
| W     |    |           |    |    |    |    |    |           |    |    |           |    |    |    |    |           |
| Reset | 0  | 0         | 0  | 0  | 0  | 0  | 0  | 0         | 0  | 0  | 0         | 0  | 0  | 0  | 0  | 0         |
| Bit   | 16 | 17        | 18 | 19 | 20 | 21 | 22 | 23        | 24 | 25 | 26        | 27 | 28 | 29 | 30 | 31        |
| R     | 0  | VTES_PRC1 |    | 0  |    |    |    | HVEN_PRC1 | 0  |    | VTES_PRC0 | 0  |    |    |    | HVEN_PRC0 |
| W     |    |           |    |    |    |    |    |           |    |    |           |    |    |    |    |           |
| Reset | 0  | 0         | 0  | 0  | 0  | 0  | 0  | 0         | 0  | 0  | 0         | 0  | 0  | 0  | 0  | 0         |

### INTC\_MCR field descriptions

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                          |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–17<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                              |
| 18<br>VTES_PRC1   | Vector Table Entry Size for Processor 1 (Core 1)<br><br>For software vector mode only, controls the number of zeroes to the right of INTVEC_PRC1 in INTC_IACKR_PRC1. If the contents of INTC_IACKR_PRC1 are used as an address of an entry in a vector table, then the number of right-most zeroes determines the size of each vector table entry.<br><br>0   4 bytes<br>1   8 bytes |
| 19–22<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                              |
| 23<br>HVEN_PRC1   | Hardware Vector Enable for Processor 1 (Core 1)<br><br>Controls whether the INTC is in hardware vector mode or software vector mode. See <a href="#">Modes of operation</a> for the details of the handshaking with the processor in each mode.<br><br>0   Software vector mode<br>1   Hardware vector mode                                                                          |
| 24–25<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                              |
| 26<br>VTES_PRC0   | Vector Table Entry Size for Processor 0 (Core 0)                                                                                                                                                                                                                                                                                                                                     |

Table continues on the next page...

**INTC\_MCR field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                    |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | For software vector mode only, controls the number of zeroes to the right of INTVEC_PRC0 in INTC_IACKR_PRC0. If the contents of INTC_IACKR_PRC0 are used as an address of an entry in a vector table, then the number of right-most zeroes determines the size of each vector table entry.<br><br>0    4 bytes<br>1    8 bytes |
| 27–30<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                        |
| 31<br>HVEN_PRC0   | Hardware Vector Enable for Processor 0 (Core 0)<br><br>Controls whether the INTC is in hardware vector mode or software vector mode. See <a href="#">Modes of operation</a> for the details of the handshaking with the processor in each mode.<br><br>0    Software vector mode<br>1    Hardware vector mode                  |

**15.7.2 INTC Current Priority Register for Processor *n* (INTC\_CPR\_PRC<sub>*n*</sub>)**

The INTC\_CPR\_PRC<sub>*n*</sub> masks any peripheral or software-settable interrupt request that is set at the same or lower priority as the current value of the INTC\_CPR\_PRC<sub>*n*</sub>[PRI] field from generating an interrupt request to the processor. When the INTC Interrupt Acknowledge register (INTC\_IACKR\_PRC<sub>*n*</sub>) is read in software vector mode or the interrupt acknowledge signal from the processor is asserted in hardware vector mode, the value of PRI is pushed onto the LIFO, and PRI is updated with the priority of the preempting interrupt request. When the INTC End-of-Interrupt register (INTC\_EOIR\_PRC<sub>*n*</sub>) is written, the LIFO is popped into the INTC\_CPR\_PRC<sub>*n*</sub>[PRI] field. An exception case in hardware vector mode to this behavior is described in [Hardware vector mode](#).

The masking priority can be raised or lowered by writing to the PRI field, supporting the priority ceiling protocol. See [Priority ceiling protocol](#).

**NOTE**

The PRI register address space must be configured as guarded and cache-inhibited prior to writing to the register.

**NOTE**

A store to modify the PRI field that closely precedes or follows an access to a shared resource may result in a non-coherent access to that resource. See [Ensuring coherency](#) for example code to ensure coherency.

Address: 0h base + 8h offset + (4d × i), where i=0d to 1d

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27  | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | PRI |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 1  | 1  | 1  | 1  |

### INTC\_CPR\_PRC<sub>n</sub> field descriptions

| Field            | Description                                                                                                               |
|------------------|---------------------------------------------------------------------------------------------------------------------------|
| 0–27<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                   |
| 28–31<br>PRI     | Priority of the currently executing ISR, according to the field values 15 (highest priority) down to 0 (lowest priority). |

## 15.7.3 INTC Interrupt Acknowledge Register for Processor *n* (INTC\_IACKR\_PRC<sub>n</sub>)

The Interrupt Acknowledge Register for Processor *n* provides a value which can be used to load the address of an ISR from a vector table. The vector table can be composed of addresses of the ISRs specific to their respective interrupt vectors.

### NOTE

Also, in software vector mode, the INTC\_IACKR\_PRC<sub>n</sub> has side effects from reads. Therefore, it must not be read speculatively while in this mode. The side effects are the same regardless of the size of the read.

### NOTE

Reading the INTC\_IACKR\_PRC<sub>n</sub> has no side effects in hardware vector mode (when the corresponding INTC\_MCR[HVEN\_PRC<sub>n</sub>] field is 1).

### NOTE

In software vector mode, the INTC\_IACKR\_PRC<sub>n</sub> must be read before setting MSR[EE]. No synchronization instruction is needed after reading the INTC\_IACKR\_PRC<sub>n</sub> and before setting MSR[EE].

However, the time for the processor to recognize the assertion or negation of the external input to it is not defined by the book E architecture and can be greater than 0. Therefore, insert instructions between the reading of the INTC\_IACKR\_PRC<sub>n</sub> and the setting of MSR[EE] that consumes at least two

processor clock cycles. This length of time allows the interrupt request negation to propagate through the processor before MSR[EE] is set.

Address: 0h base + 10h offset + (4d × i), where i=0d to 1d

|       |           |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |             |    |    |    |    |    |    |    |    |    |    |    |
|-------|-----------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|-------------|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0         | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20          | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | VTBA_PRCn |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    | INTVEC_PRCn |    |    |    |    |    |    |    | 0  |    |    |    |
| W     |           |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |             |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0         | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0           | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

### INTC\_IACKR\_PRCn field descriptions

| Field                | Description                                                                                                                                                                                                                                                                  |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–20<br>VTBA_PRCn    | Vector Table Base Address<br>Can be the base address of a vector table of addresses of ISRs. VTBA_PRCn uses only the left-most 20 bits when the corresponding VTES_PRCn bit in INTC_MCR is asserted.                                                                         |
| 21–29<br>INTVEC_PRCn | Interrupt Vector<br>Vector of the peripheral or software-settable interrupt request that caused the interrupt request to the processor. When the interrupt request to the processor asserts, the INTVEC is updated, whether the INTC is in software or hardware vector mode. |
| 30–31<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                      |

## 15.7.4 INTC End Of Interrupt Register for Processor n (INTC\_EOIR\_PRCn)

Writing to the INTC\_EOIR\_PRCn signals the end of the servicing of the interrupt request. When the INTC\_EOIR\_PRCn is written, the priority last pushed on the LIFO is popped into INTC\_CPR\_PRCn. The values and size of data written to the INTC\_EOIR\_PRCn are ignored. Those values and sizes written to this register neither update the INTC\_EOIR\_PRCn contents nor affect whether the LIFO pops. Typically, when you write to this register, write four all-zero bytes.

Address: 0h base + 18h offset + (4d × i), where i=0d to 1d

|       |          |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0        | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |          |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     | EOI_PRCn |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**INTC\_EOIR\_PRCn field descriptions**

| Field            | Description                                                                                                             |
|------------------|-------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>EOI_PRCn | End of Interrupt<br>Write four all-zero bytes to this field to signal the end of the servicing of an interrupt request. |

**15.7.5 INTC Software Set/Clear Interrupt Register (INTC\_SSCIRn)**

The INTC\_SSCIR registers support the setting or clearing of software-settable interrupt requests. These registers contain independent sets of bits to set and clear a corresponding flag bit by software. With the exception of being set by software, this flag bit behaves the same as a flag bit set within a peripheral. This flag bit generates an interrupt request within the INTC just like a peripheral interrupt request. Writing a 1 to SETn will leave SETn unchanged at 0 but will set CLRn. Writing a 0 to SETn will have no effect. CLRn is the flag bit. Writing a 1 to CLRn will clear it. Writing a 0 to CLRn will have no effect. If a 1 is written to a pair of SETn and CLRn bits at the same time, CLRn is asserted, regardless of whether CLRn was asserted before the write.

Address: 0h base + 20h offset + (1d × i), where i=0d to 7d

|       |   |   |   |   |   |   |      |      |
|-------|---|---|---|---|---|---|------|------|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6    | 7    |
| Read  | 0 |   |   |   |   |   | 0    | CLRn |
| Write |   |   |   |   |   |   | SETn | w1c  |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0    | 0    |

**INTC\_SSCIRn field descriptions**

| Field           | Description                                                                                                                                                                                                                                                                                      |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–5<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                          |
| 6<br>SETn       | Set flag bits. Writing a 1 will set the corresponding CLRn bit. Writing a 0 will have no effect. Each SETn is read as a 0.                                                                                                                                                                       |
| 7<br>CLRn       | Clear flag bits. CLRn is the flag bit. Writing a 1 to CLRn will clear it provided that a 1 is not written simultaneously to its corresponding SETn bit. Writing a 0 to CLRn will have no effect.<br><br>0 Interrupt request not pending within INTC.<br>1 Interrupt request pending within INTC. |

### 15.7.6 INTC Priority Select Register (INTC\_PSRn)

The Priority Select Registers support the selection of an individual priority for each source of interrupt request, and the routing of the request to one or both processors. The unique vector of each peripheral or software-settable interrupt request determines which INTC\_PSRn is assigned to that interrupt request. The software-settable interrupt requests 0–7 are assigned vectors 0–7, and their priorities are configured in INTC\_PSR0 to INTC\_PSR7, respectively. The peripheral interrupt requests are assigned to vectors in the range 8 to 511, and their priorities are configured in the range of registers INTC\_PSR8 to INTC\_PSR511, respectively.

**NOTE**

Interrupts are not detected while the PSRs are being written.  
The PRC\_SELn or PRIn field of an INTC\_PSRn must not be modified while the corresponding peripheral or software settable interrupt request is asserted.

Address: 0h base + 40h offset + (1d × i), where i=0d to 511d



INTC\_PSRn field descriptions

| Field           | Description                                                                                                                                                                                                                                                                                                                  |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–1<br>PRC_SELn | Processor select bits. If an interrupt source is enabled, PRC_SELn[3:0] selects whether the interrupt request is to be sent to processor 0, processor 1, or both.<br><br>00 Interrupt request sent to processor 0<br>01 Interrupt request sent to both processors<br>10 Reserved<br>11 Interrupt request sent to processor 1 |
| 2–3<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                      |
| 4–7<br>PRIn     | Priority select. Selects the priority for the interrupt requests. PRI has values 15 (highest priority) down to 0 (lowest priority).                                                                                                                                                                                          |

## 15.8 Functional description

The functional description involves the areas of interrupt request sources, priority management, and handshaking with the processor. In addition, spaces in the memory map have been reserved for other possible implementations of the INTC.



## 15.8.1 Interrupt request sources

The INTC has two types of interrupt requests, peripheral and software-settable. These interrupt requests can assert on any clock cycle.

The INTC has no spurious vector support. Therefore, if an asserted peripheral or software-settable interrupt request, whose  $PRI_n$  value in  $INTC\_PSR_n$  is higher than the  $PRI$  value in  $INTC\_CPR_n$ , negates before the interrupt request to the processor for that peripheral or software-settable interrupt request is acknowledged, then the interrupt request to the processor still can assert (or will remain asserted) for that peripheral or software-settable interrupt request. In this case, the interrupt vector will correspond to that peripheral or software-settable interrupt request. Also, the  $PRI$  value in the associated  $INTC\_CPR_n$  is updated with the corresponding  $PRI_n$  value in  $INTC\_PSR_n$ .

Furthermore, clearing the peripheral interrupt request's enable bit in the peripheral, or (alternatively) setting its mask bit, has the same consequences as clearing its flag bit. Setting its enable bit or clearing its mask bit while its flag bit is asserted has the same effect on the INTC as an interrupt event setting the flag bit.

### 15.8.1.1 Peripheral interrupt requests

An interrupt event in a peripheral's hardware sets a flag bit which resides in that peripheral. The interrupt request from the peripheral is driven by that flag bit.

The time from when the peripheral starts to drive its peripheral interrupt request to the INTC, to the time that the INTC starts to drive the interrupt request to the processor, is three clocks.

Interrupt requests from devices external to this chip are classified as peripheral interrupt requests in this document. This type of external peripheral interrupts is handled by the SIU.

### 15.8.1.2 Software-settable interrupt requests

The software set/clear interrupt registers ( $INTC\_SSCIR_n$ ) support the setting or clearing of software-settable interrupt requests. These registers contain independent sets of bits to set and clear a corresponding flag bit by software. An interrupt request is triggered by software by writing a 1 to a  $SET_n$  bit in  $INTC\_SSCIR_n$ . This write sets the corresponding  $CLR_n$  bit, which is a flag bit, resulting in the interrupt request. The interrupt request is cleared by writing a 1 to the  $CLR_n$  bit. Specific behavior includes the following:

- Writing a 1 to SET $n$  leaves SET $n$  unchanged at 0 but sets the flag bit (which is the CLR $n$  bit).
- Writing a 0 to SET $n$  has no effect.
- Writing a 1 to CLR $n$  clears the flag (CLR $n$ ) bit.
- Writing a 0 to CLR $n$  has no effect.
- If a 1 is written to a pair of SET $n$  and CLR $n$  bits at the same time, the flag (CLR $n$ ) is set, regardless of whether CLR $n$  was asserted before the write.

The time from the write to the SET $n$  bit, to the time that the INTC starts to drive the interrupt request to the processor, is four clocks.

### 15.8.1.3 Unique vector for each interrupt request source

Each peripheral and software-settable interrupt request is assigned a hardwired unique 9-bit vector. Software-settable interrupts 0–7 are assigned vectors 0–7, respectively. The peripheral interrupt requests are assigned vectors from 8 to a number as high as needed to cover all peripheral interrupt requests.

## 15.8.2 Priority management

The asserted interrupt requests are compared to each other based on their PRI $n$  and PRC\_SEL $n$  values set in INTC\_PSR $n$ . The result of that comparison also is compared to PRI in the associated INTC\_CPR $n$ . The results of those comparisons are used to manage the priority of the ISR being executed by the associated processor. The associated LIFO also assists in managing that priority.

### 15.8.2.1 Current priority and preemption

The priority arbitrator, selector, encoder, and comparator logic shown in [Figure 15-1](#) are used to compare the priority of the asserted interrupt requests to the current priority. If the priority of any asserted peripheral or software-settable interrupt request is higher than the current priority for a given processor, then the interrupt request to the processor is asserted. Also, a unique vector for the preempting peripheral or software-settable interrupt request is generated for the associated INTC\_IACKR $n$ , and if in hardware vector mode, for the interrupt vector provided to the processor.

### 15.8.2.1.1 Priority arbitrator

The priority arbitrator for each processor compares all the priorities of all of the asserted interrupt requests assigned to that processor, both peripheral and software-settable. The output of the priority arbitrator is the highest of those priorities assigned to a given processor. Also, any interrupt requests which have this highest priority are output as asserted interrupt requests to the associated selector logic.

### 15.8.2.1.2 Selector

If only one interrupt request from the associated priority arbitrator is asserted, then it is passed as asserted to the associated encoder logic. If multiple interrupt requests from the associated priority arbitrator are asserted, then only the one with the lowest vector is passed as asserted to the associated encoder logic. The lower vector is chosen regardless of the time order of the assertions of the peripheral or software-settable interrupt requests.

### 15.8.2.1.3 Encoder

The encoder logic generates the unique 9-bit vector for the asserted interrupt request from the request selector submodule for the associated processor.

### 15.8.2.1.4 Comparator

The comparator logic compares the highest priority output from the associated priority arbitrator subblock with PRI in the associated INTC\_CPR $n$ . If the comparator detects that this highest priority is higher than the current priority, then it asserts the interrupt request to the associated processor. This interrupt request to the processor asserts whether this highest priority is raised above the value of PRI in the associated INTC\_CPR $n$ , or the PRI value in the associated INTC\_CPR $n$  is lowered below this highest priority. This highest priority then becomes the new priority which is written to PRI in the associated INTC\_CPR $n$  when the interrupt request to the processor is acknowledged. Interrupt requests whose PRI $n$  in INTC\_PSR $n$  are zero will not cause a preemption because their PRI $n$  will not be higher than PRI in the associated INTC\_CPR $n$ .

Another function of the comparator is to signal an update of the INTC\_IACKR $n$  with the vector number of the first interrupt that arrives that has a priority higher than the current priority. Once the vector number and priority are captured, they cannot be superseded by a higher priority interrupt until an update of the INTC\_CPR $n$  occurs.

One consequence of the priority comparator design is that once a higher priority interrupt is captured, it must be acknowledged by the CPU before a subsequent interrupt request of even higher priority can be captured. For example, if the CPU is executing a priority level

1 interrupt, and a priority level 2 interrupt request is captured by the INTC, followed shortly by a priority level 3 interrupt request to the INTC, the level 2 interrupt must be acknowledged by the CPU before a new level 3 interrupt will be generated.

### 15.8.2.2 Stack (LIFO)

The LIFO stores the preempted PRI values from the associated INTC\_CPR $n$ . Therefore, because these priorities are stacked within the INTC, if interrupts need to be enabled during the ISR, at the beginning of the interrupt exception handler the PRI value in the associated INTC\_CPR $n$  does not need to be loaded from the associated INTC\_CPR $n$  and stored onto the context stack. Likewise at the end of the interrupt exception handler, the priority does not need to be loaded from the context stack and stored into the associated INTC\_CPR $n$ .

The PRI value in the associated INTC\_CPR $n$  is pushed onto the LIFO when the associated INTC\_IACKR $n$  is read in software vector mode or the interrupt acknowledge signal from the associated processor is asserted in hardware vector mode. The priority is popped into PRI in the associated INTC\_CPR $n$  whenever the associated INTC\_EOIR $n$  is written. An exception case in hardware vector mode to this behavior is described in [Hardware vector mode](#).

Although the INTC supports 16 priorities, an ISR executing with PRI in the INTC\_CPR $n$  equal to 15 will not be preempted. Therefore, the LIFO supports the stacking of 15 priorities. However, the LIFO is only 14 entries deep. An entry for a priority of 0 is not needed because of the ways that pushing onto a full LIFO and popping an empty LIFO are treated. If the LIFO is pushed 15 or more times than it is popped, the priorities first pushed are overwritten. A priority of 0 would be an overwritten priority. However, the LIFO will pop zeroes if it is popped more times than it is pushed. Therefore, although a priority of 0 was overwritten, it is regenerated with the popping of an empty LIFO.

## 15.8.3 Handshaking with processor

This section describes handshaking with the processor.

### 15.8.3.1 Software vector mode handshaking

This section describes the software vector mode handshaking.

### 15.8.3.1.1 Acknowledging interrupt request to processor

A timing diagram of the interrupt request and acknowledge handshaking in software vector mode, along with the handshaking near the end of the interrupt exception handler, is shown in [Figure 15-539](#). The INTC examines the peripheral and software-settable interrupt requests. When it finds an asserted peripheral or software-settable interrupt request with a higher priority than PRI in the associated INTC\_CPR<sub>n</sub>, it asserts the interrupt request to the associated processor. The INTVEC field in the associated INTC\_IACKR<sub>n</sub> is updated with the preempting interrupt request's vector when the interrupt request to the processor is asserted. The INTVEC field retains that value until the next time the interrupt request to the processor is asserted. The rest of the handshaking is described in [Software vector mode](#).

### 15.8.3.1.2 End of interrupt exception handler

Before the interrupt exception handling completes, INTC\_EOIR\_PRC<sub>n</sub> must be written. When it is written, the associated LIFO is popped so that the preempted priority is restored into PRI of the associated INTC\_CPR\_PRC<sub>n</sub>. Before it is written, the peripheral or software-settable flag bit must be cleared so that the peripheral or software-settable interrupt request is negated.

#### NOTE

To ensure proper operation across all Power Architecture MCUs, execute an **mbar** or **msync** instruction between the access to clear the flag bit and the write to the INTC\_EOIR\_PRC<sub>n</sub>.

When returning from the preemption, the INTC does not search for the peripheral or software-settable interrupt request whose ISR was preempted. Depending on how much the ISR has progressed, that interrupt request may no longer even be asserted. When PRI in the associated INTC\_CPR\_PRC<sub>n</sub> is lowered to the priority of the preempted ISR, the interrupt request for the preempted ISR or any other asserted peripheral or software-settable interrupt request at or below that priority will not cause a preemption. Instead, after the restoration of the preempted context, the processor will return to the instruction address that it had been going to execute next, before it was preempted. This next instruction is part of the preempted ISR or the interrupt exception handler's prolog or epilog.

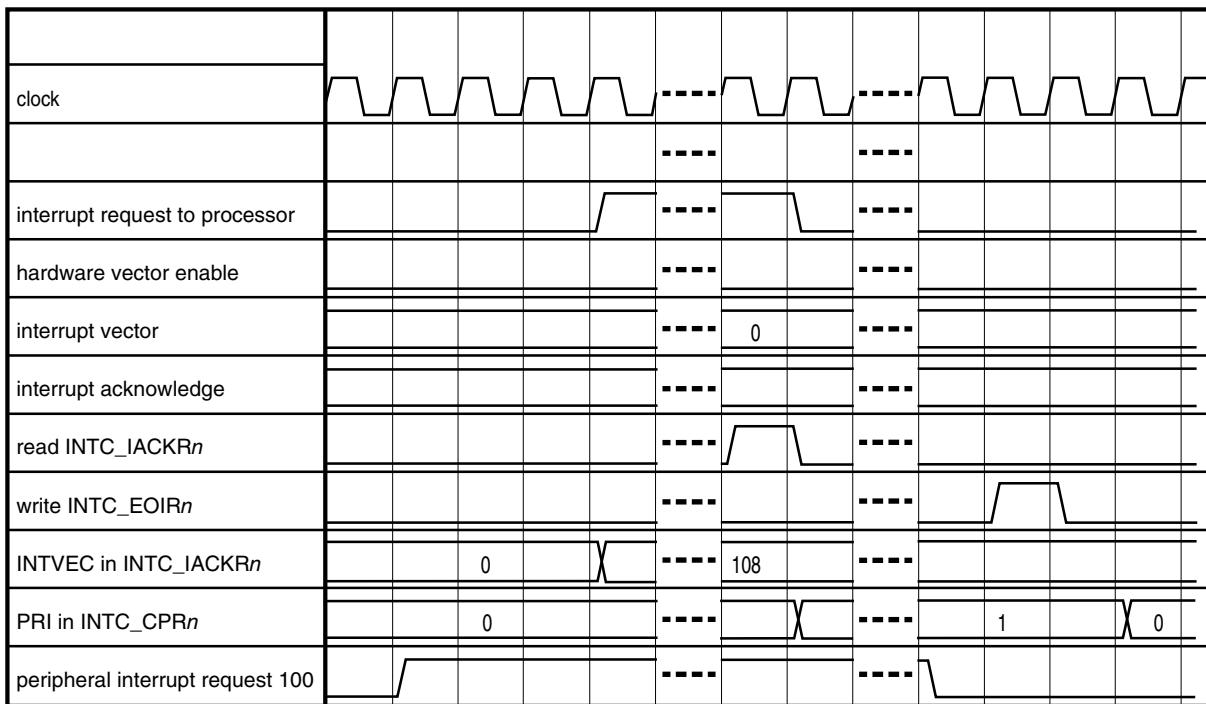


**Figure 15-539. Timing diagram of software vector mode handshaking**

### 15.8.3.2 Hardware vector mode handshaking

A timing diagram of the interrupt request and acknowledge handshaking in hardware vector mode, along with the handshaking near the end of the interrupt exception handler, is shown in [Figure 15-540](#). As in software vector mode, the INTC examines the peripheral and software-settable interrupt requests, and when it finds an asserted interrupt request with a higher priority than PRI in the associated INTC\_CPR $n$ , it asserts the interrupt request to the associated processor. The INTVEC field in the associated INTC\_IACKR $n$  is updated with the preempting peripheral or software-settable interrupt request's vector when the interrupt request to the processor is asserted. The INTVEC field retains that value until the next time the interrupt request to the associated processor is asserted. In addition, the value of the interrupt vector to the associated processor matches the value of the INTVEC field in the associated INTC\_IACKR $n$ . The rest of the handshaking is described in [Hardware vector mode](#).

The handshaking near the end of the interrupt exception handler, that is the writing to the associated `INTC_EOIRn`, is the same as in software vector mode. See [End of interrupt exception handler](#).

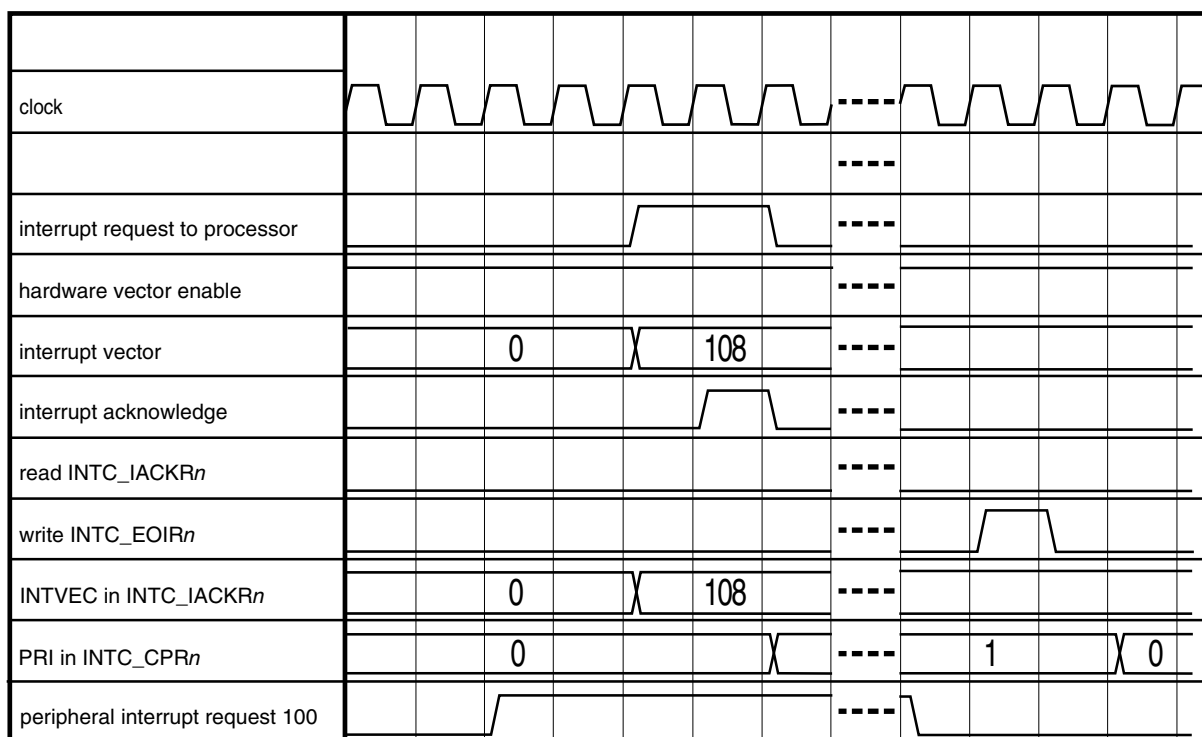


**Figure 15-540. Timing diagram for hardware vector mode handshaking**

## 15.9 Initialization/application information

This section contains initialization/application information.

### 15.9.1 Initialization flow

After exiting reset, all of the  $PRI_n$  and  $PRC\_SEL_n$  fields in  $INTC\_PSR_n$  are zero, and  $PRI$  in the  $INTC\_CPR_n$  registers is 15d. These reset values prevent the INTC from asserting the interrupt request to the processors. The enable or mask bits in the peripherals are reset such that the peripheral interrupt requests are negated. An initialization sequence for allowing the peripheral and software-settable interrupt requests to cause an interrupt request to the processor is:

- interrupt\_request\_initialization:
  - Configure  $HVEN\_PRC_n$  in  $INTC\_MCR$ .
  - Configure  $VTBA\_PRC_n$  in  $INTC\_IACKR\_PRC_n$ .
  - Raise the  $PRI_n$  fields and set the  $PRC\_SEL_n$  fields to the desired processor in  $INTC\_PSR_n$ .
  - Set the enable bits or clear the mask bits for the peripheral interrupt requests.

- Lower PRI in INTC\_CPR\_PRCn to zero.
- Enable recognition of interrupts by the processor(s).

## 15.9.2 Interrupt exception handler

These example interrupt exception handlers excerpts use Power Architecture assembly code.

### 15.9.2.1 Software vector mode

```

interrupt_exception_handler:
b      interrupt_exception_handler_continued# 16 bytes available, branch to continue

interrupt_exception_handler_continued:
code to create stack frame, save working register, and save SRR0 and SRR1 (Power
Architecture Save/Restore registers) (not shown)

lis      r3,INTC_IACKR_PRCn@ha      # form adjusted upper half of INTC_IACKR_PRCn address
lwz      r3,INTC_IACKR_PRCn@l(r3)   # load INTC_IACKR_PRCn, which clears request to
processor
lwz      r3,0x0(r3)                 # load address of ISR from vector table
wrteei   1                          # enable processor recognition of interrupts

code to save rest of context required by Power Architecture EABI (Embedded Binary
Application Interface) (not shown)

mtlrr    r3                          # move the INTC_IACKRn address into the link register
blrl     # branch to ISR; link register updated with epilog
# address

epilog:
code to restore most of context required by Power Architecture EABI (not shown)
# Popping the LIFO after the restoration of most of the context and the disabling of
processor
# recognition of interrupts eases the calculation of the maximum stack depth at the cost of
# postponing the servicing of the next interrupt request.
mbar     # ensure store to clear flag bit has completed
lis      r3,INTC_EOIR_PRCn@ha      # form adjusted upper half of INTC_EOIR_PRCn address
li       r4,0x0                    # form 0 to write to INTC_EOIR_PRCn
wrteei   0                          # disable processor recognition of interrupts
stw      r4,INTC_EOIR_PRCn@l(r3)   # store to INTC_EOIR_PRCn, informing INTC to lower
priority
code to restore SRR0 and SRR1, restore working registers, and delete stack frame (not shown)

rfi

vector_table_base_address:
address of ISR for interrupt with vector 0
address of ISR for interrupt with vector 1
.
.
.
address of ISR for interrupt with vector 510
address of ISR for interrupt with vector 511
ISRx:
code to service the interrupt event (not shown)
code to clear flag bit which drives interrupt request to INTC (not shown)
blr     # return to epilog

```



### 15.9.2.2 Hardware vector mode

This interrupt exception handler is useful with processor and system bus implementations that support a hardware vector. This example assumes that each

`interrupt_exception_handlerx` only has space for four instructions, and therefore a branch to `interrupt_exception_handler_continuedx` is needed.

```

interrupt_exception_handlerx:
b   interrupt_exception_handler_continuedx# 16 bytes available, branch to continue

interrupt_exception_handler_continuedx:
code to create stack frame, save working register, and save SRR0 and SRR1 (not shown)

wrteei 1      # enable processor recognition of interrupts

code to save rest of context required by Power Architecture EABI (not shown)

bl   ISRx     # branch to ISR for interrupt with vector x

epilog:
code to restore most of context required by Power Architecture EABI (not shown)

# Popping the LIFO after the restoration of most of the context and the disabling of
processor
# recognition of interrupts eases the calculation of the maximum stack depth at the cost of
# postponing the servicing of the next interrupt request.
mbar          # ensure store to clear flag bit has completed
lis   r3,INTC_EOIR_PRCn@ha # form adjusted upper half of INTC_EOIR_PRCn address
li    r4,0x0    # form 0 to write to INTC_EOIR_PRCn
wrteei 0      # disable processor recognition of interrupts
stw   r4,INTC_EOIR_PRCn@l(r3) # store to INTC_EOIR_PRCn, informing INTC to lower priority

code to restore SRR0 and SRR1, restore working registers, and delete stack frame (not shown)

rfi

ISRx:
code to service the interrupt event
code to clear flag bit which drives interrupt request to INTC

blr   # branch to epilog

```

### 15.9.3 ISR, RTOS, and task hierarchy

The RTOS and all of the tasks under its control typically execute with PRI in `INTC_CPRn` having a value of 0. The RTOS will execute the tasks according to whatever priority scheme it may have, but that priority scheme is independent and has a lower priority of execution than the priority scheme of the INTC. In other words, the ISRs execute above `INTC_CPRn` priority 0 and outside the control of the RTOS, the RTOS executes at `INTC_CPRn` priority 0, and while the tasks execute at different priorities under the control of the RTOS, they also execute at `INTC_CPRn` priority 0.

If a task shares a resource with an ISR and the Priority Ceiling Protocol (PCP) is being used to manage that shared resource, then the task's priority can be elevated in the `INTC_CPRn` while the shared resource is being accessed.

An ISR whose `PRIn` in `INTC_PSRn` has a value of 0 will not cause an interrupt request to the selected processor, even if its peripheral or software-settable interrupt request is asserted. For a peripheral interrupt request, not setting its enable bit or disabling the mask bit will cause it to remain negated, which consequently also will not cause an interrupt request to the processor. Since the ISRs are outside the control of the RTOS, this ISR will not run unless called by another ISR or the interrupt exception handler, perhaps after executing another ISR.

### 15.9.4 Order of execution

An ISR with a higher priority can preempt an ISR with a lower priority, regardless of the unique vectors associated with each of their peripheral or software-settable interrupt requests. However, if multiple peripheral or software-settable interrupt requests are asserted, more than one of these interrupt requests has the highest priority and that priority is high enough to cause preemption, the INTC selects the one with the lowest unique vector regardless of the order in time that they asserted. However, the ability to meet deadlines with this scheduling scheme is no less than if the ISRs execute in the time order that their peripheral or software-settable interrupt requests asserted.

The example in the following table shows the order of execution of both cases, ISRs with different priorities and ISRs with the same priority.

**Table 15-534. Order of ISR execution example**

| Step # | Step Description                                                                             | Code executing at end of step |                     |        |        |        |                             | PRI in INTC_CPR at end of step |
|--------|----------------------------------------------------------------------------------------------|-------------------------------|---------------------|--------|--------|--------|-----------------------------|--------------------------------|
|        |                                                                                              | RTOS                          | ISR108 <sub>1</sub> | ISR208 | ISR308 | ISR408 | interrupt exception handler |                                |
| 1      | RTOS at priority 0 is executing.                                                             | X                             |                     |        |        |        |                             | 0                              |
| 2      | Peripheral interrupt request 100 at priority 1 asserts. Interrupt taken.                     |                               | X                   |        |        |        |                             | 1                              |
| 3      | Peripheral interrupt request 400 at priority 4 asserts. Interrupt taken.                     |                               |                     |        |        | X      |                             | 4                              |
| 4      | Peripheral interrupt request 300 at priority 3 asserts.                                      |                               |                     |        |        | X      |                             | 4                              |
| 5      | Peripheral interrupt request 200 at priority 3 asserts.                                      |                               |                     |        |        | X      |                             | 4                              |
| 6      | ISR408 completes. Interrupt exception handler writes to <code>INTC_EOIR<sub>n</sub></code> . |                               |                     |        |        |        | X                           | 1                              |

*Table continues on the next page...*

**Table 15-534. Order of ISR execution example (continued)**

| Step # | Step Description                                                                                        | Code executing at end of step |                     |        |        |        |                             | PRI in INTC_CPR at end of step |
|--------|---------------------------------------------------------------------------------------------------------|-------------------------------|---------------------|--------|--------|--------|-----------------------------|--------------------------------|
|        |                                                                                                         | RTOS                          | ISR108 <sup>1</sup> | ISR208 | ISR308 | ISR408 | interrupt exception handler |                                |
| 7      | Interrupt taken. ISR208 starts to execute, even though peripheral interrupt request 300 asserted first. |                               |                     | X      |        |        |                             | 3                              |
| 8      | ISR208 completes. Interrupt exception handler writes to INTC_EOIR <sub>n</sub> .                        |                               |                     |        |        |        | X                           | 1                              |
| 9      | Interrupt taken. ISR308 starts to execute.                                                              |                               |                     |        | X      |        |                             | 3                              |
| 10     | ISR308 completes. Interrupt exception handler writes to INTC_EOIR <sub>n</sub> .                        |                               |                     |        |        |        | X                           | 1                              |
| 11     | ISR108 completes. Interrupt exception handler writes to INTC_EOIR <sub>n</sub> .                        |                               |                     |        |        |        | X                           | 0                              |
| 12     | RTOS continues execution.                                                                               | X                             |                     |        |        |        |                             | 0                              |

1. ISR108 executes for peripheral interrupt request 100 because the first eight ISRs are for software-settable interrupt requests.

## 15.9.5 Priority ceiling protocol

This section describes the priority ceiling protocol.

### 15.9.5.1 Elevating priority

The PRI field in INTC current priority register (INTC\_CPR<sub>n</sub>) is elevated in the OSEK PCP to the ceiling of all of the priorities of the ISRs that share a resource. This protocol therefore allows coherent accesses of the ISRs to that shared resource.

For example, ISR1 has a priority of 1, ISR2 has a priority of 2, and ISR3 has a priority of 3. They all share the same resource. Before ISR1 or ISR2 can access that resource, they must raise the PRI value in INTC\_CPR<sub>n</sub> to 3, the ceiling of all of the ISR priorities. After they release the resource, they must lower the PRI value in INTC\_CPR<sub>n</sub> to prevent further priority inversion. If they do not raise their priority, then ISR2 can preempt ISR1, and ISR3 can preempt ISR1 or ISR2, possibly corrupting the shared resource. Another possible failure mechanism is deadlock if the higher priority ISR needs the lower priority ISR to release the resource before it can continue, but the lower priority ISR cannot release the resource until the higher priority ISR completes and execution returns to the lower priority ISR.

Using the PCP instead of disabling processor recognition of all interrupts eliminates the time when accessing a shared resource that all higher priority interrupts are blocked. For example, while ISR3 cannot preempt ISR1 while it is accessing the shared resource, all of the ISRs with a priority higher than 3 can preempt ISR1.

### 15.9.5.2 Ensuring coherency

This section discusses how to ensure coherency.

#### 15.9.5.2.1 Interrupt with blocked priority

A scenario can exist that can cause non-coherent accesses to the shared resource. As an example, ISR1 and ISR2 both share a resource. ISR1 has a lower priority than ISR2. ISR1 is executing, and it writes to the `INTC_CPRn`. The instruction following this store is a store to a value in a shared coherent data block. Either just before or at the same time as the first store, the INTC asserts the interrupt request to the processor because the peripheral interrupt request for ISR2 has asserted. As the processor is responding to the interrupt request from the INTC, and as it is aborting transactions and flushing its pipeline, it is possible (in some implementations) that both of these stores are executed. ISR2 thereby assumes that it can access the data block coherently, but the data block has been corrupted.

When the Current Priority Register's (INTC\_CPR\_PCR's) PRI field in the INTC is updated by software, external interrupts from the INTC to the core can be disabled so that an external interrupt will never be taken in the window of time when a system resource (PRI field) is being changed. Disabling interrupts and re-enabling them after the PRI elevation, as well as providing additional core clocks of delay after interrupts have been enabled, provides enough delay for the ISR that has elevated its PRI to be interrupted by a pending ISR before the ISR elevating PRI accesses a shared memory block.

To ensure coherent access to the shared data block, modifications to PRI in INTC\_CPR\_PRCx can be made by services managing shared resources with the following code sequence:

- disable processor recognition of interrupts
- PRI modification (guarded, cache inhibited)
- enable processor recognition of interrupts
- delay 5 core clocks
- coherent data access

The delay of 5 clocks minimum can be achieved by the following series of instructions:

```
or r1, r1, r1
```

```

or r1, r1, r1
or r1, r1, r1
or r1, r1, r1
or r1, r1, r1

```

The `or r1, r1, r1` is not dual issued because the destination of one “or” instruction is the source of the next, hence the instructions cannot be executed in parallel.

### 15.9.5.2.2 Raised priority preserved

The following figure shows the timing diagram for a scenario where a priority that is raised in one ISR is preserved after execution of a pre-empting ISR. The table following the figure explains the events. The example is for software vector mode. Except for the method of retrieving the vector and acknowledging the interrupt request to the processor, hardware vector mode is identical.

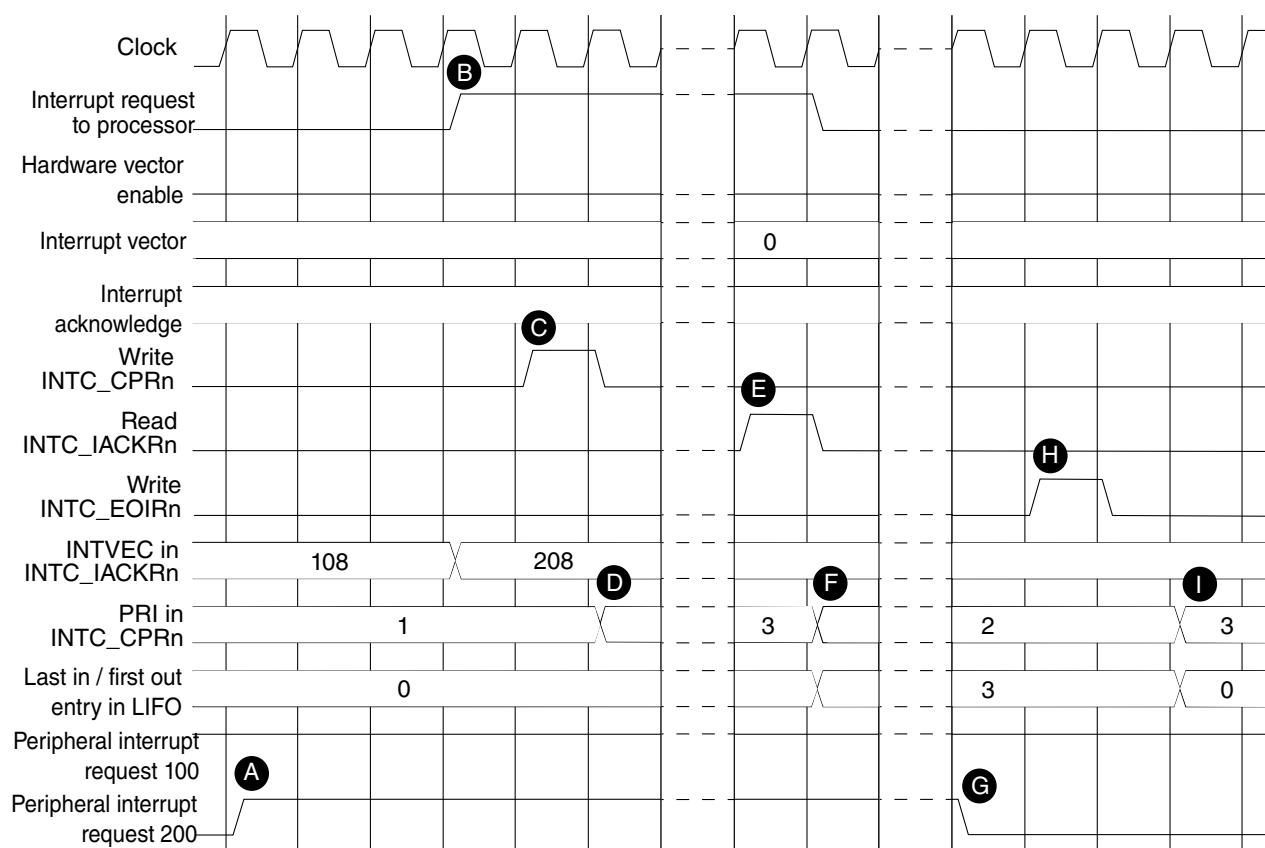


**Figure 15-541. Timing diagram of raised priority preserved**

**Table 15-535. Raised priority preserved events**

| Event | Description                                                                                                              |
|-------|--------------------------------------------------------------------------------------------------------------------------|
| A     | Peripheral interrupt request 200 asserts during execution of ISR108 running at priority 1.                               |
| B     | Interrupt request to processor asserts. INTVEC in INTC_IACKRn updates with vector for that peripheral interrupt request. |

*Table continues on the next page...*

**Table 15-535. Raised priority preserved events (continued)**

| Event | Description                                                                                                                                                                                                                                                                                  |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| C     | ISR108 writes to INTC_CPR <sub>n</sub> to raise priority to 3 before accessing shared coherent data block.                                                                                                                                                                                   |
| D     | PRI in INTC_CPR <sub>n</sub> now at 3, reflecting the write. This write, just before accessing data block, is the last instruction the processor executes before being interrupted.                                                                                                          |
| E     | Interrupt exception handler prolog acknowledges interrupt by reading INTC_IACKR <sub>n</sub> .                                                                                                                                                                                               |
| F     | PRI of 3 pushed onto LIFO. PRI in INTC_CPR <sub>n</sub> updates to 2, the priority of ISR208.                                                                                                                                                                                                |
| G     | ISR208 clears its flag bit, deasserting its peripheral interrupt request.                                                                                                                                                                                                                    |
| H     | Interrupt exception handler epilog writes to INTC_EOIR <sub>n</sub> .                                                                                                                                                                                                                        |
| I     | LIFO pops 3, restoring the raised priority onto PRI in INTC_CPR <sub>n</sub> . Next value to pop from LIFO is the priority from before peripheral interrupt request 100 interrupted. ISR108 now can access data block coherently after interrupt exception handler executes rfi instruction. |

### 15.9.6 Selecting priorities according to request rates and deadlines

The selection of the priorities for the ISRs can be made using Rate Monotonic Scheduling (RMS) or a superset of it, Deadline Monotonic Scheduling (DMS). In RMS, the ISRs which have higher request rates have higher priorities. In DMS, if the deadline is before the next time the ISR is requested, then the ISR is assigned a priority according to the time from the request for the ISR to the deadline, not from the time of the request for the ISR to the next request for it.

For example, ISR1 executes every 100  $\mu$ s, ISR2 executes every 200  $\mu$ s, and ISR3 executes every 300  $\mu$ s. ISR1 has a higher priority than ISR2 which has a higher priority than ISR3. However, if ISR3 has a deadline of 150  $\mu$ s, then it has a higher priority than ISR2.

The INTC supports 16 priorities, which may be many fewer than the number of ISRs. In this case, the ISRs should be grouped with other ISRs that have similar deadlines. For example, a priority could be allocated for every time the request rate doubles. ISRs with request rates around 1 ms would share a priority, ISRs with request rates around 500  $\mu$ s would share a priority, ISRs with request rates around 250  $\mu$ s would share a priority, and so on. With this approach, a range of ISR request rates of  $2^{16}$  could be covered, regardless of the number of ISRs.

Reducing the number of priorities does cause some priority inversion, which reduces the processor's ability to meet its deadlines. However, reducing the number of priorities can reduce the size and latency through the interrupt controller. It also allows easier management of ISRs with similar deadlines that share a resource. They can be placed at the same priority without any further priority inversion, and they do not need to use the PCP to access the shared resource.

## 15.9.7 Software-settable interrupt requests

The software-settable interrupt requests can be used in two ways. They can be used to schedule a lower priority portion of an ISR and for processors to interrupt other processors in a multiple processor system.

### 15.9.7.1 Scheduling a lower priority portion of an ISR

A portion of an ISR needs to be executed at the  $PRI_n$  value in  $INTC\_PSR_n$ , which becomes the  $PRI$  value in  $INTC\_CPR_n$  with the interrupt acknowledge. The ISR, however, can have a portion of it which does not need to be executed at this higher priority. Therefore, executing this later portion which does not need to be executed at this higher priority can prevent the execution of ISRs which do not have a higher priority than the earlier portion of the ISR but do have a higher priority than the later portion of the ISR needs. This preemptive scheduling inefficiency reduces the processor's ability to meet its deadlines.

One option is for the ISR to complete the earlier higher priority portion, but then schedule through the RTOS a task to execute the later lower priority portion. However, some RTOSs can require a large amount of time for an ISR to schedule a task. Therefore, a second option is for the ISR, after completing the higher priority portion, to set a  $SET_n$  bit in  $INTC\_SSCIR_n$ . Writing a '1' to  $SET_n$  causes a software-settable interrupt request. This software-settable interrupt request, which usually will have a lower  $PRI_n$  value in the  $INTC\_PSR_n$ , therefore will not cause preemptive scheduling inefficiencies.

After generating a software settable interrupt request, the higher priority ISR completes. The lower priority ISR is scheduled according to its priority. Execution of the higher priority ISR is not resumed after the completion of the lower priority ISR.

### 15.9.7.2 Scheduling an ISR on another processor

Since the  $SET_n$  bits in the  $INTC\_SSCIR_n$  are memory mapped, processors in multiple processor systems can schedule ISRs on the other processors. One possible application is if one processor simply wants to command another processor to perform a piece of work, and the initiating processor does not need to use the results of that work. If the initiating processor needs to know if the processor executing the software-settable ISR has not completed the work before asking it to again execute that ISR, it can check if the corresponding  $CLR_n$  bit in  $INTC\_SSCIR_n$  is asserted before again writing a 1 to the  $SET_n$  bit.

Another application is the sharing of a block of data. For example, a first processor has completed accessing a block of data and wants a second processor to then access it. Furthermore, after the second processor has completed accessing the block of data, the first processor again wants to access it. The accesses to the block of data must be done coherently. The procedure is that the first processor writes a 1 to a SET $n$  bit on the second processor. The second processor, after accessing the block of data, clears the corresponding CLR $n$  bit and then writes 1 to a SET $n$  bit on the first processor, informing it that it now can access the block of data.

### 15.9.8 Lowering priority within an ISR

In implementations without the software-settable interrupt requests in INTC\_SSCIR $n$ , one way (besides scheduling a task through an RTOS) to prevent preemptive scheduling inefficiencies with an ISR whose work spans multiple priorities (as described in [Scheduling a lower priority portion of an ISR](#)) is to lower the current priority. However, the INTC has a LIFO whose depth is determined by the number of priorities.

#### Note

Lowering the PRI value in INTC\_CPR $n$  within an ISR to below the ISR's corresponding PRI value in INTC\_PSR $n$  allows more preemptions than the depth of the LIFO can support.

Therefore, through its use of the LIFO the INTC does not support lowering the current priority within an ISR as a way to avoid preemptive scheduling inefficiencies.

### 15.9.9 Negating an interrupt request outside of its ISR

This section discusses the negation of an interrupt service request outside of its ISR.

#### 15.9.9.1 Negating an interrupt request as a side effect of an ISR

Some peripherals have flag bits which can be cleared as a side effect of servicing a peripheral interrupt request. For example, reading a specific register can clear the flag bits, and consequently their corresponding interrupt requests too. This clearing as a side effect of servicing a peripheral interrupt request can cause the negation of other peripheral interrupt requests besides the peripheral interrupt request whose ISR presently is executing. This negating of a peripheral interrupt request outside of its ISR can be a desired effect.



### 15.9.9.2 Negating multiple interrupt requests in one ISR

An ISR can clear other flag bits besides its own flag bit. One reason that an ISR clears multiple flag bits is because it serviced those other flag bits, and therefore the ISRs for these other flag bits do not need to be executed.

### 15.9.9.3 Proper setting of interrupt request priority

Whether an interrupt request negates outside of its own ISR due to the side effect of an ISR execution or the intentional clearing of a flag bit, the priorities of the peripheral or software-settable interrupt requests for these other flag bits must be selected properly. Their  $PRI_n$  values in  $INTC\_PSR_n$  must be selected to be at or lower than the priority of the ISR that cleared their flag bits. Otherwise, those flag bits still can cause the interrupt request to the processor to assert. Furthermore, the clearing of these other flag bits also has the same timing relationship to the writing to  $INTC\_SSCIR_n$  as the clearing of the flag bit that caused the present ISR to be executed. Refer to [End of interrupt exception handler](#) for more information.

A flag bit whose enable bit or mask bit is negating its peripheral interrupt request can be cleared at any time, regardless of the peripheral interrupt request's  $PRI_n$  value in  $INTC\_PSR_n$ .

### 15.9.10 Examining LIFO contents

Normally the user does not need to know the contents of the LIFO, or even how deep the LIFO is nested. Although the LIFO contents are not memory-mapped, the user can read the contents by popping the LIFO and reading the  $PRI$  field in the INTC current priority register ( $INTC\_CPR_n$ ). Disabling processor recognition of interrupts while examining the LIFO contents provides a coherent view of the preempted priorities. The code sequence is:

```
pop_lifo:
    store to INTC_EOIR_PRCn
    load INTC_CPR_PRCn, examine PRI, and store onto stack
    if PRI is not zero or value when interrupts were enabled, branch to pop_lifo
```

When the examination is complete, the LIFO can be restored in software vector mode using this code sequence:

```
push_lifo:
    load stacked PRI value and store to INTC_CPR_PRCn
```

```
load INTC_IACKR_PRCn  
if stacked PRI values are not depleted, branch to push_lifo
```

### Note

Reading the INTC\_IACKR\_PRC $n$  acknowledges the interrupt request to the processor and updates the INTC\_CPR\_PRC $n$ [PRI] with the priority of the preempting interrupt request. If the processor recognition of interrupts is disabled during the LIFO restoration, interrupt requests to the processor can go undetected. However, because the peripheral or software-settable interrupt requests are not cleared, the peripheral interrupt request to the processor re-asserts when INTC\_CPR\_PRC $n$ [PRI] is lower than the priorities of those peripheral or software-settable interrupt requests.

## 15.10 Interrupt sources

See [Interrupt vector assignments](#) for the chip-specific list of interrupt sources.

# Chapter 16

## Direct Memory Access Controller (eDMA)

### 16.1 Chip-specific eDMA information

The chip has two instances of the eDMA module: eDMA\_A and eDMA\_B. Each instance supports 64 channels.

See [DMA request mapping](#) for the channel allocation of DMA requests for each module instance.

The eDMA has no direct external MCU signals. However, four external pins can be configured in the SIU to route external eDMA requests to four eDMA\_A channels. See [DMA/Interrupt Request Select Register \(SIU\\_DIRSR\)](#) for details.

### 16.2 Introduction

The enhanced direct memory access (eDMA) controller is a second-generation module capable of performing complex data transfers with minimal intervention from a host processor. The hardware microarchitecture includes:

- A DMA engine that performs:
  - Source address and destination address calculations
  - Data-movement operations
- Local memory containing transfer control descriptors for each of the 64 channels

#### 16.2.1 eDMA system block diagram

[Figure 16-1](#) illustrates the components of the eDMA system, including the eDMA module ("engine").

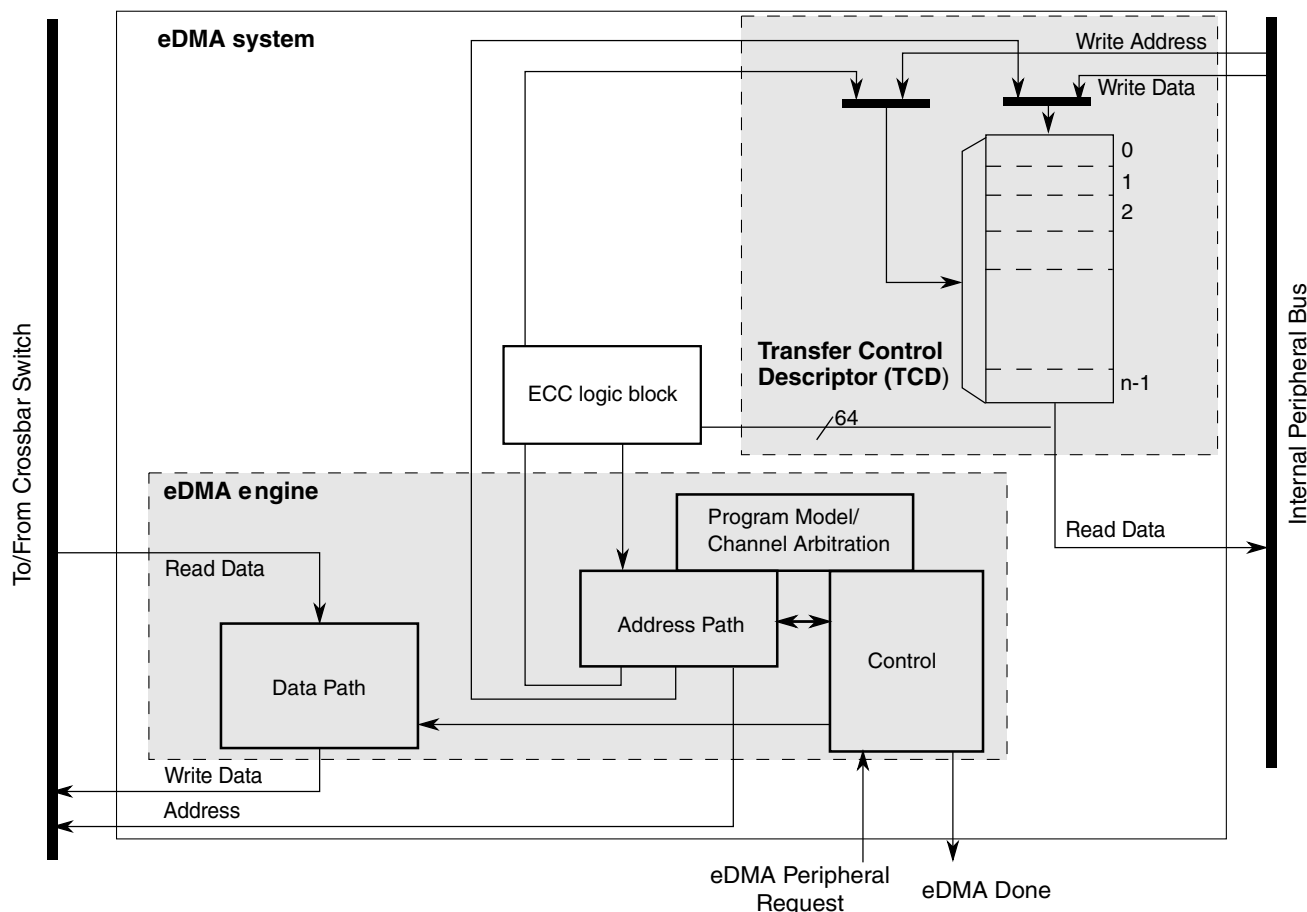


Figure 16-1. eDMA system block diagram

16.2.2 Block parts

The eDMA module is partitioned into two major modules: the eDMA engine and the transfer-control descriptor local memory.

The eDMA engine is further partitioned into four submodules:

Table 16-1. eDMA engine submodules

| Submodule    | Function                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Address path | <p>This block implements registered versions of two channel transfer control descriptors, channel x and channel y, and manages all master bus-address calculations. All the channels provide the same functionality. This structure allows data transfers associated with one channel to be preempted after the completion of a read/write sequence if a higher priority channel activation is asserted while the first channel is active. After a channel is activated, it runs until the minor loop is completed, unless preempted by a higher priority channel. This provides a mechanism (enabled by DCHPRI<sub>n</sub>[ECP]) where a large data move operation can be preempted to minimize the time another channel is blocked from execution.</p> <p>When any channel is selected to execute, the contents of its TCD are read from local memory and loaded into the address path channel x registers for a normal start and into channel y registers for a preemption start. After the minor loop completes execution, the address path hardware writes</p> |

Table continues on the next page...

**Table 16-1. eDMA engine submodules (continued)**

| Submodule                         | Function                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                   | the new values for the TCDn_{SADDR, DADDR, CITER} back to local memory. If the major iteration count is exhausted, additional processing is performed, including the final address pointer updates, reloading the TCDn_CITER field, and a possible fetch of the next TCDn from memory as part of a scatter/gather operation.                                                                                                                                                                                                                                                                 |
| Data path                         | <p>This block implements the bus master read/write datapath. It includes a data buffer and the necessary multiplex logic to support any required data alignment. The internal read data bus is the primary input, and the internal write data bus is the primary output.</p> <p>The address and data path modules directly support the 2-stage pipelined internal bus. The address path module represents the 1st stage of the bus pipeline (address phase), while the data path module implements the 2nd stage of the pipeline (data phase).</p>                                           |
| Program model/channel arbitration | This block implements the first section of the eDMA programming model as well as the channel arbitration logic. The programming model registers are connected to the internal peripheral bus. The eDMA peripheral request inputs and interrupt request outputs are also connected to this block (via control logic).                                                                                                                                                                                                                                                                         |
| Control                           | This block provides all the control functions for the eDMA engine. For data transfers where the source and destination sizes are equal, the eDMA engine performs a series of source read/destination write operations until the number of bytes specified in the minor loop byte count has moved. For descriptors where the sizes are not equal, multiple accesses of the smaller size data are required for each reference of the larger size. As an example, if the source size references 16-bit data and the destination is 32-bit data, two reads are performed, then one 32-bit write. |

The transfer-control descriptor local memory is further partitioned into:

**Table 16-2. Transfer control descriptor memory**

| Submodule         | Description                                                                                                                                                                                                                                                                                     |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Memory controller | This logic implements the required dual-ported controller, managing accesses from the eDMA engine as well as references from the internal peripheral bus. As noted earlier, in the event of simultaneous accesses, the eDMA engine is given priority and the peripheral transaction is stalled. |
| Memory array      | TCD storage for each channel's transfer profile.                                                                                                                                                                                                                                                |

### 16.2.3 Features

The eDMA is a highly programmable data-transfer engine optimized to minimize any required intervention from the host processor. It is intended for use in applications where the data size to be transferred is statically known and not defined within the transferred data itself. The eDMA module features:

- All data movement via dual-address transfers: read from source, write to destination
  - Programmable source and destination addresses and transfer size
  - Support for enhanced addressing modes

- 64-channel implementation that performs complex data transfers with minimal intervention from a host processor
  - Internal data buffer, used as temporary storage to support 16- and 32-byte transfers
  - Connections to the crossbar switch for bus mastering the data movement
- Transfer control descriptor (TCD) organized to support two-deep, nested transfer operations
  - 32-byte TCD stored in local memory for each channel
  - An inner data transfer loop defined by a minor byte transfer count
  - An outer data transfer loop defined by a major iteration count
- Channel activation via one of three methods:
  - Explicit software initiation
  - Initiation via a channel-to-channel linking mechanism for continuous transfers
  - Peripheral-paced hardware requests, one per channel
- Fixed-priority and round-robin channel arbitration
- Channel completion reported via programmable interrupt requests
  - One interrupt per channel, which can be asserted at completion of major iteration count
  - Programmable error terminations per channel and logically summed together to form one error interrupt to the interrupt controller
- Programmable support for scatter/gather DMA processing
- Support for complex data structures
- Error detection and error correction

In the discussion of this module,  $n$  is used to reference the channel number.

## 16.3 Modes of operation

The eDMA operates in the following modes:

**Table 16-3. Modes of operation**

| Mode   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Normal | <p>In Normal mode, the eDMA transfers data between a source and a destination. The source and destination can be a memory block or an I/O block capable of operation with the eDMA.</p> <p>A service request initiates a transfer of a specific number of bytes (NBYTES) as specified in the transfer control descriptor (TCD). The minor loop is the sequence of read-write operations that transfers these NBYTES per service request. Each service request executes one iteration of the major loop, which transfers NBYTES of data.</p> |
| Debug  | <p>DMA operation is configurable in Debug mode via the control register:</p> <ul style="list-style-type: none"> <li>• If CR[EDBG] is cleared, the DMA continues to operate.</li> <li>• If CR[EDBG] is set, the eDMA stops transferring data. If Debug mode is entered while a channel is active, the eDMA continues operation until the channel retires.</li> </ul>                                                                                                                                                                         |
| Wait   | Before entering Wait mode, the DMA attempts to complete its current transfer. After the transfer completes, the device enters Wait mode.                                                                                                                                                                                                                                                                                                                                                                                                    |

## 16.4 Memory map/register definition

The eDMA's programming model is partitioned into two regions:

- The first region defines a number of registers providing control functions
- The second region corresponds to the local transfer control descriptor (TCD) memory

### 16.4.1 TCD memory

Each channel requires a 32-byte transfer control descriptor for defining the desired data movement operation. The channel descriptors are stored in the local memory in sequential order: channel 0, channel 1, ... channel 63. Each TCD<sub>*n*</sub> definition is presented as 11 registers of 16 or 32 bits.

Some registers are implemented as two 32-bit registers, and include an H and L suffix, signaling the high and low portions of the control function. This table provides a 32-bit view of the eDMA's memory map.

## 16.4.2 TCD initialization

The initialization of the TCD memory is performed by the eDMA controller after reset. The initialization runs for 256 eDMA clock cycles. All fields in the TCD memory are initialized to 0. All application read or write accesses to the TCD are delayed until the initialization is finished. Prior to activating a channel, you must initialize its TCD with the appropriate transfer profile.

## 16.4.3 TCD structure

### DMA Basics: TCD Structure

- One DMA engine has a number of channels to react to DMA requests
- Each channel has its own TCD

| Word<br>Offset | 0                   | 1                     | 2                            | 3 | 4     | 5 | 6 | 7     | 8    | 9 | 10 | 11 | 12    | 13 | 14 | 15   | 16   | 17           | 18 | 19 | 20 | 21 | 22                  | 23   | 24     | 25           | 26   | 27    | 28       | 29      | 30    | 31 |
|----------------|---------------------|-----------------------|------------------------------|---|-------|---|---|-------|------|---|----|----|-------|----|----|------|------|--------------|----|----|----|----|---------------------|------|--------|--------------|------|-------|----------|---------|-------|----|
| 0x1000         | SADDR               |                       |                              |   |       |   |   |       |      |   |    |    |       |    |    |      |      |              |    |    |    |    |                     |      |        |              |      |       |          |         |       |    |
| 0x1004         | SMOD                |                       |                              |   | SSIZE |   |   |       | DMOD |   |    |    | DSIZE |    |    |      | SOFF |              |    |    |    |    |                     |      |        |              |      |       |          |         |       |    |
| 0x1008         | NBYTES <sup>1</sup> |                       |                              |   |       |   |   |       |      |   |    |    |       |    |    |      |      |              |    |    |    |    |                     |      |        |              |      |       |          |         |       |    |
| 0x1008         | SMLOE <sup>1</sup>  | DMLOE <sup>1</sup>    | MLOFF or NBYTES <sup>1</sup> |   |       |   |   |       |      |   |    |    |       |    |    |      |      |              |    |    |    |    | NBYTES <sup>1</sup> |      |        |              |      |       |          |         |       |    |
| 0x100C         | SLAST               |                       |                              |   |       |   |   |       |      |   |    |    |       |    |    |      |      |              |    |    |    |    |                     |      |        |              |      |       |          |         |       |    |
| 0x1010         | DADDR               |                       |                              |   |       |   |   |       |      |   |    |    |       |    |    |      |      |              |    |    |    |    |                     |      |        |              |      |       |          |         |       |    |
| 0x1014         | CITER.E_LINK        | CITER or CITER.LINKCH |                              |   |       |   |   | CITER |      |   |    |    |       |    |    | DOFF |      |              |    |    |    |    |                     |      |        |              |      |       |          |         |       |    |
| 0x1018         | DLAST_SGA           |                       |                              |   |       |   |   |       |      |   |    |    |       |    |    |      |      |              |    |    |    |    |                     |      |        |              |      |       |          |         |       |    |
| 0x101C         | BITER.E_LINK        | BITER or BITER.LINKCH |                              |   |       |   |   | BITER |      |   |    |    |       |    |    | BWC  |      | MAJOR LINKCH |    |    |    |    |                     | DONE | ACTIVE | MAJOR.E_LINK | E_SG | D_REQ | INT_HALF | INT_MAJ | START |    |
|                | 0                   | 1                     | 2                            | 3 | 4     | 5 | 6 | 7     | 8    | 9 | 10 | 11 | 12    | 13 | 14 | 15   | 16   | 17           | 18 | 19 | 20 | 21 | 22                  | 23   | 24     | 25           | 26   | 27    | 28       | 29      | 30    | 31 |

<sup>1</sup> The fields implemented in Word 2 depend on whether EDMA\_CR[EMLM] bit is set to 0 or 1.

## 16.4.4 Reserved memory and bit fields

- Reading reserved bits in a register returns the value of zero.



- Writes to reserved bits in a register are ignored.
- Reading or writing a reserved memory location generates a bus error.

## DMA memory map

| Address offset (hex) | Register name                                    | Width (in bits) | Access                | Reset value | Section/ page               |
|----------------------|--------------------------------------------------|-----------------|-----------------------|-------------|-----------------------------|
| 0                    | Control Register (DMA_CR)                        | 32              | R/W                   | 0000_E400h  | <a href="#">16.4.5/623</a>  |
| 4                    | Error Status Register (DMA_ES)                   | 32              | R                     | 0000_0000h  | <a href="#">16.4.6/626</a>  |
| 8                    | Enable Request Register High (DMA_ERQH)          | 32              | R/W                   | 0000_0000h  | <a href="#">16.4.7/629</a>  |
| C                    | Enable Request Register Low (DMA_ERQL)           | 32              | R/W                   | 0000_0000h  | <a href="#">16.4.8/632</a>  |
| 10                   | Enable Error Interrupt Register High (DMA_EEIH)  | 32              | R/W                   | 0000_0000h  | <a href="#">16.4.9/636</a>  |
| 14                   | Enable Error Interrupt Register Low (DMA_EEIL)   | 32              | R/W                   | 0000_0000h  | <a href="#">16.4.10/639</a> |
| 18                   | Set Enable Request Register (DMA_SERQ)           | 8               | W<br>(always reads 0) | 00h         | <a href="#">16.4.11/643</a> |
| 19                   | Clear Enable Request Register (DMA_CERQ)         | 8               | W<br>(always reads 0) | 00h         | <a href="#">16.4.12/643</a> |
| 1A                   | Set Enable Error Interrupt Register (DMA_SEEI)   | 8               | W<br>(always reads 0) | 00h         | <a href="#">16.4.13/644</a> |
| 1B                   | Clear Enable Error Interrupt Register (DMA_CEEI) | 8               | W<br>(always reads 0) | 00h         | <a href="#">16.4.14/645</a> |
| 1C                   | Clear Interrupt Request Register (DMA_CINT)      | 8               | W<br>(always reads 0) | 00h         | <a href="#">16.4.15/646</a> |
| 1D                   | Clear Error Register (DMA_CERR)                  | 8               | W<br>(always reads 0) | 00h         | <a href="#">16.4.16/647</a> |
| 1E                   | Set START Bit Register (DMA_SSRT)                | 8               | W<br>(always reads 0) | 00h         | <a href="#">16.4.17/648</a> |
| 1F                   | Clear DONE Status Bit Register (DMA_CDNE)        | 8               | W<br>(always reads 0) | 00h         | <a href="#">16.4.18/649</a> |
| 20                   | Interrupt Request Register High (DMA_INTH)       | 32              | R/W                   | 0000_0000h  | <a href="#">16.4.19/649</a> |
| 24                   | Interrupt Request Register Low (DMA_INTL)        | 32              | R/W                   | 0000_0000h  | <a href="#">16.4.20/653</a> |
| 28                   | Error Register High (DMA_ERRH)                   | 32              | R/W                   | 0000_0000h  | <a href="#">16.4.21/657</a> |
| 2C                   | Error Register Low (DMA_ERRL)                    | 32              | R/W                   | 0000_0000h  | <a href="#">16.4.22/661</a> |
| 30                   | Hardware Request Status Register High (DMA_HRSH) | 32              | R                     | 0000_0000h  | <a href="#">16.4.23/664</a> |
| 34                   | Hardware Request Status Register Low (DMA_HRSL)  | 32              | R                     | 0000_0000h  | <a href="#">16.4.24/668</a> |

Table continues on the next page...

## DMA memory map (continued)

| Address offset (hex) | Register name                              | Width (in bits) | Access | Reset value                 | Section/page                |
|----------------------|--------------------------------------------|-----------------|--------|-----------------------------|-----------------------------|
| 38                   | Global PCU Output Register (DMA_GPORH)     | 32              | R/W    | 0000_0000h                  | <a href="#">16.4.25/672</a> |
| 3C                   | Global PCU Output Register (DMA_GPORL)     | 32              | R/W    | 0000_0000h                  | <a href="#">16.4.25/672</a> |
| 100                  | Channel n Priority Register (DMA_DCHPRI0)  | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/673</a> |
| 101                  | Channel n Priority Register (DMA_DCHPRI1)  | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/673</a> |
| 102                  | Channel n Priority Register (DMA_DCHPRI2)  | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/673</a> |
| 103                  | Channel n Priority Register (DMA_DCHPRI3)  | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/673</a> |
| 104                  | Channel n Priority Register (DMA_DCHPRI4)  | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/673</a> |
| 105                  | Channel n Priority Register (DMA_DCHPRI5)  | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/673</a> |
| 106                  | Channel n Priority Register (DMA_DCHPRI6)  | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/673</a> |
| 107                  | Channel n Priority Register (DMA_DCHPRI7)  | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/673</a> |
| 108                  | Channel n Priority Register (DMA_DCHPRI8)  | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/673</a> |
| 109                  | Channel n Priority Register (DMA_DCHPRI9)  | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/673</a> |
| 10A                  | Channel n Priority Register (DMA_DCHPRI10) | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/673</a> |
| 10B                  | Channel n Priority Register (DMA_DCHPRI11) | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/673</a> |
| 10C                  | Channel n Priority Register (DMA_DCHPRI12) | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/673</a> |
| 10D                  | Channel n Priority Register (DMA_DCHPRI13) | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/673</a> |
| 10E                  | Channel n Priority Register (DMA_DCHPRI14) | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/673</a> |
| 10F                  | Channel n Priority Register (DMA_DCHPRI15) | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/673</a> |
| 110                  | Channel n Priority Register (DMA_DCHPRI16) | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/673</a> |
| 111                  | Channel n Priority Register (DMA_DCHPRI17) | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/673</a> |
| 112                  | Channel n Priority Register (DMA_DCHPRI18) | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/673</a> |
| 113                  | Channel n Priority Register (DMA_DCHPRI19) | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/673</a> |

Table continues on the next page...

**DMA memory map (continued)**

| Address offset (hex) | Register name                              | Width (in bits) | Access | Reset value                 | Section/ page                |
|----------------------|--------------------------------------------|-----------------|--------|-----------------------------|------------------------------|
| 114                  | Channel n Priority Register (DMA_DCHPRI20) | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/ 673</a> |
| 115                  | Channel n Priority Register (DMA_DCHPRI21) | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/ 673</a> |
| 116                  | Channel n Priority Register (DMA_DCHPRI22) | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/ 673</a> |
| 117                  | Channel n Priority Register (DMA_DCHPRI23) | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/ 673</a> |
| 118                  | Channel n Priority Register (DMA_DCHPRI24) | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/ 673</a> |
| 119                  | Channel n Priority Register (DMA_DCHPRI25) | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/ 673</a> |
| 11A                  | Channel n Priority Register (DMA_DCHPRI26) | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/ 673</a> |
| 11B                  | Channel n Priority Register (DMA_DCHPRI27) | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/ 673</a> |
| 11C                  | Channel n Priority Register (DMA_DCHPRI28) | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/ 673</a> |
| 11D                  | Channel n Priority Register (DMA_DCHPRI29) | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/ 673</a> |
| 11E                  | Channel n Priority Register (DMA_DCHPRI30) | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/ 673</a> |
| 11F                  | Channel n Priority Register (DMA_DCHPRI31) | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/ 673</a> |
| 120                  | Channel n Priority Register (DMA_DCHPRI32) | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/ 673</a> |
| 121                  | Channel n Priority Register (DMA_DCHPRI33) | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/ 673</a> |
| 122                  | Channel n Priority Register (DMA_DCHPRI34) | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/ 673</a> |
| 123                  | Channel n Priority Register (DMA_DCHPRI35) | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/ 673</a> |
| 124                  | Channel n Priority Register (DMA_DCHPRI36) | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/ 673</a> |
| 125                  | Channel n Priority Register (DMA_DCHPRI37) | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/ 673</a> |
| 126                  | Channel n Priority Register (DMA_DCHPRI38) | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/ 673</a> |
| 127                  | Channel n Priority Register (DMA_DCHPRI39) | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/ 673</a> |
| 128                  | Channel n Priority Register (DMA_DCHPRI40) | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/ 673</a> |
| 129                  | Channel n Priority Register (DMA_DCHPRI41) | 8               | R/W    | <a href="#">See section</a> | <a href="#">16.4.26/ 673</a> |

*Table continues on the next page...*

## DMA memory map (continued)

| Address offset (hex) | Register name                              | Width (in bits) | Access | Reset value | Section/page    |
|----------------------|--------------------------------------------|-----------------|--------|-------------|-----------------|
| 12A                  | Channel n Priority Register (DMA_DCHPRI42) | 8               | R/W    | See section | 16.4.26/<br>673 |
| 12B                  | Channel n Priority Register (DMA_DCHPRI43) | 8               | R/W    | See section | 16.4.26/<br>673 |
| 12C                  | Channel n Priority Register (DMA_DCHPRI44) | 8               | R/W    | See section | 16.4.26/<br>673 |
| 12D                  | Channel n Priority Register (DMA_DCHPRI45) | 8               | R/W    | See section | 16.4.26/<br>673 |
| 12E                  | Channel n Priority Register (DMA_DCHPRI46) | 8               | R/W    | See section | 16.4.26/<br>673 |
| 12F                  | Channel n Priority Register (DMA_DCHPRI47) | 8               | R/W    | See section | 16.4.26/<br>673 |
| 130                  | Channel n Priority Register (DMA_DCHPRI48) | 8               | R/W    | See section | 16.4.26/<br>673 |
| 131                  | Channel n Priority Register (DMA_DCHPRI49) | 8               | R/W    | See section | 16.4.26/<br>673 |
| 132                  | Channel n Priority Register (DMA_DCHPRI50) | 8               | R/W    | See section | 16.4.26/<br>673 |
| 133                  | Channel n Priority Register (DMA_DCHPRI51) | 8               | R/W    | See section | 16.4.26/<br>673 |
| 134                  | Channel n Priority Register (DMA_DCHPRI52) | 8               | R/W    | See section | 16.4.26/<br>673 |
| 135                  | Channel n Priority Register (DMA_DCHPRI53) | 8               | R/W    | See section | 16.4.26/<br>673 |
| 136                  | Channel n Priority Register (DMA_DCHPRI54) | 8               | R/W    | See section | 16.4.26/<br>673 |
| 137                  | Channel n Priority Register (DMA_DCHPRI55) | 8               | R/W    | See section | 16.4.26/<br>673 |
| 138                  | Channel n Priority Register (DMA_DCHPRI56) | 8               | R/W    | See section | 16.4.26/<br>673 |
| 139                  | Channel n Priority Register (DMA_DCHPRI57) | 8               | R/W    | See section | 16.4.26/<br>673 |
| 13A                  | Channel n Priority Register (DMA_DCHPRI58) | 8               | R/W    | See section | 16.4.26/<br>673 |
| 13B                  | Channel n Priority Register (DMA_DCHPRI59) | 8               | R/W    | See section | 16.4.26/<br>673 |
| 13C                  | Channel n Priority Register (DMA_DCHPRI60) | 8               | R/W    | See section | 16.4.26/<br>673 |
| 13D                  | Channel n Priority Register (DMA_DCHPRI61) | 8               | R/W    | See section | 16.4.26/<br>673 |
| 13E                  | Channel n Priority Register (DMA_DCHPRI62) | 8               | R/W    | See section | 16.4.26/<br>673 |
| 13F                  | Channel n Priority Register (DMA_DCHPRI63) | 8               | R/W    | See section | 16.4.26/<br>673 |

Table continues on the next page...

**DMA memory map (continued)**

| Address offset (hex) | Register name                                                                                           | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|---------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 1000                 | TCD Source Address (DMA_TCD0_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 1004                 | TCD Transfer Attributes (DMA_TCD0_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 1006                 | TCD Signed Source Address Offset (DMA_TCD0_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 1008                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD0_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 1008                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD0_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 1008                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD0_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 100C                 | TCD Last Source Address Adjustment (DMA_TCD0_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 1010                 | TCD Destination Address (DMA_TCD0_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 1014                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD0_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 1014                 | DMA_TCD0_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 1016                 | TCD Signed Destination Address Offset (DMA_TCD0_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 1018                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD0_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 101C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD0_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 101C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD0_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 101E                 | TCD Control and Status (DMA_TCD0_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 1020                 | TCD Source Address (DMA_TCD1_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 1024                 | TCD Transfer Attributes (DMA_TCD1_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 1026                 | TCD Signed Source Address Offset (DMA_TCD1_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 1028                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD1_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 1028                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD1_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 1028                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD1_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |

*Table continues on the next page...*

## DMA memory map (continued)

| Address offset (hex) | Register name                                                                                           | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|---------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 102C                 | TCD Last Source Address Adjustment (DMA_TCD1_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 1030                 | TCD Destination Address (DMA_TCD1_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 1034                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD1_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 1034                 | DMA_TCD1_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 1036                 | TCD Signed Destination Address Offset (DMA_TCD1_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 1038                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD1_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 103C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD1_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 103C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD1_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 103E                 | TCD Control and Status (DMA_TCD1_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 1040                 | TCD Source Address (DMA_TCD2_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 1044                 | TCD Transfer Attributes (DMA_TCD2_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 1046                 | TCD Signed Source Address Offset (DMA_TCD2_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 1048                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD2_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 1048                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD2_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 1048                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD2_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 104C                 | TCD Last Source Address Adjustment (DMA_TCD2_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 1050                 | TCD Destination Address (DMA_TCD2_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 1054                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD2_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 1054                 | DMA_TCD2_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 1056                 | TCD Signed Destination Address Offset (DMA_TCD2_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 1058                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD2_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |

Table continues on the next page...

**DMA memory map (continued)**

| Address offset (hex) | Register name                                                                                           | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|---------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 105C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD2_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 105C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD2_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 105E                 | TCD Control and Status (DMA_TCD2_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 1060                 | TCD Source Address (DMA_TCD3_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 1064                 | TCD Transfer Attributes (DMA_TCD3_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 1066                 | TCD Signed Source Address Offset (DMA_TCD3_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 1068                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD3_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 1068                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD3_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 1068                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD3_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 106C                 | TCD Last Source Address Adjustment (DMA_TCD3_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 1070                 | TCD Destination Address (DMA_TCD3_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 1074                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD3_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 1074                 | DMA_TCD3_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 1076                 | TCD Signed Destination Address Offset (DMA_TCD3_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 1078                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD3_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 107C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD3_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 107C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD3_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 107E                 | TCD Control and Status (DMA_TCD3_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 1080                 | TCD Source Address (DMA_TCD4_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 1084                 | TCD Transfer Attributes (DMA_TCD4_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 1086                 | TCD Signed Source Address Offset (DMA_TCD4_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |

Table continues on the next page...



## DMA memory map (continued)

| Address offset (hex) | Register name                                                                                           | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|---------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 1088                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD4_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 1088                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD4_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 1088                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD4_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 108C                 | TCD Last Source Address Adjustment (DMA_TCD4_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 1090                 | TCD Destination Address (DMA_TCD4_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 1094                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD4_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 1094                 | DMA_TCD4_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 1096                 | TCD Signed Destination Address Offset (DMA_TCD4_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 1098                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD4_DLASTGA)                       | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 109C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD4_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 109C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD4_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 109E                 | TCD Control and Status (DMA_TCD4_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 10A0                 | TCD Source Address (DMA_TCD5_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 10A4                 | TCD Transfer Attributes (DMA_TCD5_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 10A6                 | TCD Signed Source Address Offset (DMA_TCD5_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 10A8                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD5_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 10A8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD5_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 10A8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD5_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 10AC                 | TCD Last Source Address Adjustment (DMA_TCD5_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 10B0                 | TCD Destination Address (DMA_TCD5_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 10B4                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD5_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |

Table continues on the next page...



## DMA memory map (continued)

| Address offset (hex) | Register name                                                                                           | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|---------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 10B4                 | DMA_TCD5_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 10B6                 | TCD Signed Destination Address Offset (DMA_TCD5_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 10B8                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD5_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 10BC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD5_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 10BC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD5_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 10BE                 | TCD Control and Status (DMA_TCD5_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 10C0                 | TCD Source Address (DMA_TCD6_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 10C4                 | TCD Transfer Attributes (DMA_TCD6_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 10C6                 | TCD Signed Source Address Offset (DMA_TCD6_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 10C8                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD6_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 10C8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD6_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 10C8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD6_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 10CC                 | TCD Last Source Address Adjustment (DMA_TCD6_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 10D0                 | TCD Destination Address (DMA_TCD6_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 10D4                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD6_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 10D4                 | DMA_TCD6_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 10D6                 | TCD Signed Destination Address Offset (DMA_TCD6_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 10D8                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD6_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 10DC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD6_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 10DC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD6_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 10DE                 | TCD Control and Status (DMA_TCD6_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |

Table continues on the next page...

## DMA memory map (continued)

| Address offset (hex) | Register name                                                                                           | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|---------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 10E0                 | TCD Source Address (DMA_TCD7_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 10E4                 | TCD Transfer Attributes (DMA_TCD7_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 10E6                 | TCD Signed Source Address Offset (DMA_TCD7_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 10E8                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD7_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 10E8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD7_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 10E8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD7_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 10EC                 | TCD Last Source Address Adjustment (DMA_TCD7_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 10F0                 | TCD Destination Address (DMA_TCD7_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 10F4                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD7_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 10F4                 | DMA_TCD7_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 10F6                 | TCD Signed Destination Address Offset (DMA_TCD7_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 10F8                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD7_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 10FC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD7_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 10FC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD7_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 10FE                 | TCD Control and Status (DMA_TCD7_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 1100                 | TCD Source Address (DMA_TCD8_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 1104                 | TCD Transfer Attributes (DMA_TCD8_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 1106                 | TCD Signed Source Address Offset (DMA_TCD8_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 1108                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD8_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 1108                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD8_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 1108                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD8_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |

Table continues on the next page...

**DMA memory map (continued)**

| Address offset (hex) | Register name                                                                                           | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|---------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 110C                 | TCD Last Source Address Adjustment (DMA_TCD8_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 1110                 | TCD Destination Address (DMA_TCD8_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 1114                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD8_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 1114                 | DMA_TCD8_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 1116                 | TCD Signed Destination Address Offset (DMA_TCD8_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 1118                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD8_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 111C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD8_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 111C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD8_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 111E                 | TCD Control and Status (DMA_TCD8_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 1120                 | TCD Source Address (DMA_TCD9_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 1124                 | TCD Transfer Attributes (DMA_TCD9_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 1126                 | TCD Signed Source Address Offset (DMA_TCD9_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 1128                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD9_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 1128                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD9_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 1128                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD9_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 112C                 | TCD Last Source Address Adjustment (DMA_TCD9_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 1130                 | TCD Destination Address (DMA_TCD9_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 1134                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD9_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 1134                 | DMA_TCD9_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 1136                 | TCD Signed Destination Address Offset (DMA_TCD9_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 1138                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD9_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |

*Table continues on the next page...*

## DMA memory map (continued)

| Address offset (hex) | Register name                                                                                            | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|----------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 113C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD9_BITER_ELINKYES)      | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 113C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD9_BITER_ELINKNO)      | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 113E                 | TCD Control and Status (DMA_TCD9_CSR)                                                                    | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 1140                 | TCD Source Address (DMA_TCD10_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 1144                 | TCD Transfer Attributes (DMA_TCD10_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 1146                 | TCD Signed Source Address Offset (DMA_TCD10_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 1148                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD10_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 1148                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD10_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 1148                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD10_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 114C                 | TCD Last Source Address Adjustment (DMA_TCD10_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 1150                 | TCD Destination Address (DMA_TCD10_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 1154                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD10_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 1154                 | DMA_TCD10_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 1156                 | TCD Signed Destination Address Offset (DMA_TCD10_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 1158                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD10_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 115C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD10_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 115C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD10_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 115E                 | TCD Control and Status (DMA_TCD10_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 1160                 | TCD Source Address (DMA_TCD11_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 1164                 | TCD Transfer Attributes (DMA_TCD11_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 1166                 | TCD Signed Source Address Offset (DMA_TCD11_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |

Table continues on the next page...

## DMA memory map (continued)

| Address offset (hex) | Register name                                                                                            | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|----------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 1168                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD11_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 1168                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD11_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 1168                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD11_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 116C                 | TCD Last Source Address Adjustment (DMA_TCD11_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 1170                 | TCD Destination Address (DMA_TCD11_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 1174                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD11_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 1174                 | DMA_TCD11_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 1176                 | TCD Signed Destination Address Offset (DMA_TCD11_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 1178                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD11_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 117C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD11_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 117C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD11_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 117E                 | TCD Control and Status (DMA_TCD11_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 1180                 | TCD Source Address (DMA_TCD12_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 1184                 | TCD Transfer Attributes (DMA_TCD12_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 1186                 | TCD Signed Source Address Offset (DMA_TCD12_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 1188                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD12_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 1188                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD12_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 1188                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD12_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 118C                 | TCD Last Source Address Adjustment (DMA_TCD12_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 1190                 | TCD Destination Address (DMA_TCD12_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 1194                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD12_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |

Table continues on the next page...

## DMA memory map (continued)

| Address offset (hex) | Register name                                                                                            | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|----------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 1194                 | DMA_TCD12_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 1196                 | TCD Signed Destination Address Offset (DMA_TCD12_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 1198                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD12_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 119C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD12_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 119C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD12_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 119E                 | TCD Control and Status (DMA_TCD12_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 11A0                 | TCD Source Address (DMA_TCD13_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 11A4                 | TCD Transfer Attributes (DMA_TCD13_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 11A6                 | TCD Signed Source Address Offset (DMA_TCD13_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 11A8                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD13_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 11A8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD13_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 11A8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD13_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 11AC                 | TCD Last Source Address Adjustment (DMA_TCD13_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 11B0                 | TCD Destination Address (DMA_TCD13_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 11B4                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD13_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 11B4                 | DMA_TCD13_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 11B6                 | TCD Signed Destination Address Offset (DMA_TCD13_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 11B8                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD13_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 11BC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD13_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 11BC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD13_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |

Table continues on the next page...

**DMA memory map (continued)**

| Address offset (hex) | Register name                                                                                            | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|----------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 11BE                 | TCD Control and Status (DMA_TCD13_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 11C0                 | TCD Source Address (DMA_TCD14_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 11C4                 | TCD Transfer Attributes (DMA_TCD14_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 11C6                 | TCD Signed Source Address Offset (DMA_TCD14_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 11C8                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD14_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 11C8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD14_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 11C8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD14_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 11CC                 | TCD Last Source Address Adjustment (DMA_TCD14_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 11D0                 | TCD Destination Address (DMA_TCD14_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 11D4                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD14_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 11D4                 | DMA_TCD14_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 11D6                 | TCD Signed Destination Address Offset (DMA_TCD14_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 11D8                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD14_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 11DC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD14_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 11DC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD14_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 11DE                 | TCD Control and Status (DMA_TCD14_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 11E0                 | TCD Source Address (DMA_TCD15_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 11E4                 | TCD Transfer Attributes (DMA_TCD15_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 11E6                 | TCD Signed Source Address Offset (DMA_TCD15_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 11E8                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD15_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 11E8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD15_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |

*Table continues on the next page...*



## DMA memory map (continued)

| Address offset (hex) | Register name                                                                                            | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|----------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 11E8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD15_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 11EC                 | TCD Last Source Address Adjustment (DMA_TCD15_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 11F0                 | TCD Destination Address (DMA_TCD15_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 11F4                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD15_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 11F4                 | DMA_TCD15_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 11F6                 | TCD Signed Destination Address Offset (DMA_TCD15_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 11F8                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD15_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 11FC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD15_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 11FC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD15_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 11FE                 | TCD Control and Status (DMA_TCD15_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 1200                 | TCD Source Address (DMA_TCD16_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 1204                 | TCD Transfer Attributes (DMA_TCD16_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 1206                 | TCD Signed Source Address Offset (DMA_TCD16_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 1208                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD16_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 1208                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD16_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 1208                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD16_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 120C                 | TCD Last Source Address Adjustment (DMA_TCD16_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 1210                 | TCD Destination Address (DMA_TCD16_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 1214                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD16_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 1214                 | DMA_TCD16_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 1216                 | TCD Signed Destination Address Offset (DMA_TCD16_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |

Table continues on the next page...



**DMA memory map (continued)**

| Address offset (hex) | Register name                                                                                            | Width (in bits) | Access | Reset value | Section/ page               |
|----------------------|----------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|-----------------------------|
| 1218                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD16_DLASTGA)                       | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/682</a> |
| 121C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD16_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/682</a> |
| 121C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD16_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/683</a> |
| 121E                 | TCD Control and Status (DMA_TCD16_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/684</a> |
| 1220                 | TCD Source Address (DMA_TCD17_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/674</a> |
| 1224                 | TCD Transfer Attributes (DMA_TCD17_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/674</a> |
| 1226                 | TCD Signed Source Address Offset (DMA_TCD17_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/675</a> |
| 1228                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD17_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/675</a> |
| 1228                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD17_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/676</a> |
| 1228                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD17_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/677</a> |
| 122C                 | TCD Last Source Address Adjustment (DMA_TCD17_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/678</a> |
| 1230                 | TCD Destination Address (DMA_TCD17_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/679</a> |
| 1234                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD17_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/679</a> |
| 1234                 | DMA_TCD17_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/680</a> |
| 1236                 | TCD Signed Destination Address Offset (DMA_TCD17_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/681</a> |
| 1238                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD17_DLASTGA)                       | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/682</a> |
| 123C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD17_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/682</a> |
| 123C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD17_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/683</a> |
| 123E                 | TCD Control and Status (DMA_TCD17_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/684</a> |
| 1240                 | TCD Source Address (DMA_TCD18_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/674</a> |

*Table continues on the next page...*

## DMA memory map (continued)

| Address offset (hex) | Register name                                                                                            | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|----------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 1244                 | TCD Transfer Attributes (DMA_TCD18_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 1246                 | TCD Signed Source Address Offset (DMA_TCD18_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 1248                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD18_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 1248                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD18_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 1248                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD18_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 124C                 | TCD Last Source Address Adjustment (DMA_TCD18_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 1250                 | TCD Destination Address (DMA_TCD18_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 1254                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD18_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 1254                 | DMA_TCD18_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 1256                 | TCD Signed Destination Address Offset (DMA_TCD18_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 1258                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD18_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 125C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD18_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 125C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD18_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 125E                 | TCD Control and Status (DMA_TCD18_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 1260                 | TCD Source Address (DMA_TCD19_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 1264                 | TCD Transfer Attributes (DMA_TCD19_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 1266                 | TCD Signed Source Address Offset (DMA_TCD19_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 1268                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD19_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 1268                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD19_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 1268                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD19_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 126C                 | TCD Last Source Address Adjustment (DMA_TCD19_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |

Table continues on the next page...

**DMA memory map (continued)**

| Address offset (hex) | Register name                                                                                            | Width (in bits) | Access | Reset value | Section/ page               |
|----------------------|----------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|-----------------------------|
| 1270                 | TCD Destination Address (DMA_TCD19_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/679</a> |
| 1274                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD19_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/679</a> |
| 1274                 | DMA_TCD19_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/680</a> |
| 1276                 | TCD Signed Destination Address Offset (DMA_TCD19_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/681</a> |
| 1278                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD19_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/682</a> |
| 127C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD19_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/682</a> |
| 127C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD19_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/683</a> |
| 127E                 | TCD Control and Status (DMA_TCD19_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/684</a> |
| 1280                 | TCD Source Address (DMA_TCD20_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/674</a> |
| 1284                 | TCD Transfer Attributes (DMA_TCD20_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/674</a> |
| 1286                 | TCD Signed Source Address Offset (DMA_TCD20_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/675</a> |
| 1288                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD20_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/675</a> |
| 1288                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD20_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/676</a> |
| 1288                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD20_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/677</a> |
| 128C                 | TCD Last Source Address Adjustment (DMA_TCD20_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/678</a> |
| 1290                 | TCD Destination Address (DMA_TCD20_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/679</a> |
| 1294                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD20_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/679</a> |
| 1294                 | DMA_TCD20_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/680</a> |
| 1296                 | TCD Signed Destination Address Offset (DMA_TCD20_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/681</a> |
| 1298                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD20_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/682</a> |
| 129C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD20_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/682</a> |

*Table continues on the next page...*

## DMA memory map (continued)

| Address offset (hex) | Register name                                                                                            | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|----------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 129C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD20_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 129E                 | TCD Control and Status (DMA_TCD20_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 12A0                 | TCD Source Address (DMA_TCD21_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 12A4                 | TCD Transfer Attributes (DMA_TCD21_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 12A6                 | TCD Signed Source Address Offset (DMA_TCD21_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 12A8                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD21_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 12A8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD21_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 12A8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD21_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 12AC                 | TCD Last Source Address Adjustment (DMA_TCD21_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 12B0                 | TCD Destination Address (DMA_TCD21_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 12B4                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD21_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 12B4                 | DMA_TCD21_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 12B6                 | TCD Signed Destination Address Offset (DMA_TCD21_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 12B8                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD21_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 12BC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD21_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 12BC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD21_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 12BE                 | TCD Control and Status (DMA_TCD21_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 12C0                 | TCD Source Address (DMA_TCD22_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 12C4                 | TCD Transfer Attributes (DMA_TCD22_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 12C6                 | TCD Signed Source Address Offset (DMA_TCD22_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 12C8                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD22_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |

Table continues on the next page...

**DMA memory map (continued)**

| Address offset (hex) | Register name                                                                                            | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|----------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 12C8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD22_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 12C8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD22_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 12CC                 | TCD Last Source Address Adjustment (DMA_TCD22_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 12D0                 | TCD Destination Address (DMA_TCD22_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 12D4                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD22_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 12D4                 | DMA_TCD22_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 12D6                 | TCD Signed Destination Address Offset (DMA_TCD22_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 12D8                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD22_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 12DC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD22_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 12DC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD22_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 12DE                 | TCD Control and Status (DMA_TCD22_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 12E0                 | TCD Source Address (DMA_TCD23_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 12E4                 | TCD Transfer Attributes (DMA_TCD23_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 12E6                 | TCD Signed Source Address Offset (DMA_TCD23_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 12E8                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD23_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 12E8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD23_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 12E8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD23_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 12EC                 | TCD Last Source Address Adjustment (DMA_TCD23_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 12F0                 | TCD Destination Address (DMA_TCD23_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 12F4                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD23_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 12F4                 | DMA_TCD23_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |

*Table continues on the next page...*

## DMA memory map (continued)

| Address offset (hex) | Register name                                                                                            | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|----------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 12F6                 | TCD Signed Destination Address Offset (DMA_TCD23_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 12F8                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD23_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 12FC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD23_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 12FC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD23_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 12FE                 | TCD Control and Status (DMA_TCD23_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 1300                 | TCD Source Address (DMA_TCD24_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 1304                 | TCD Transfer Attributes (DMA_TCD24_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 1306                 | TCD Signed Source Address Offset (DMA_TCD24_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 1308                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD24_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 1308                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD24_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 1308                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD24_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 130C                 | TCD Last Source Address Adjustment (DMA_TCD24_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 1310                 | TCD Destination Address (DMA_TCD24_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 1314                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD24_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 1314                 | DMA_TCD24_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 1316                 | TCD Signed Destination Address Offset (DMA_TCD24_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 1318                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD24_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 131C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD24_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 131C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD24_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 131E                 | TCD Control and Status (DMA_TCD24_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |

Table continues on the next page...



## DMA memory map (continued)

| Address offset (hex) | Register name                                                                                            | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|----------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 1320                 | TCD Source Address (DMA_TCD25_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 1324                 | TCD Transfer Attributes (DMA_TCD25_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 1326                 | TCD Signed Source Address Offset (DMA_TCD25_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 1328                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD25_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 1328                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD25_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 1328                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD25_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 132C                 | TCD Last Source Address Adjustment (DMA_TCD25_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 1330                 | TCD Destination Address (DMA_TCD25_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 1334                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD25_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 1334                 | DMA_TCD25_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 1336                 | TCD Signed Destination Address Offset (DMA_TCD25_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 1338                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD25_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 133C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD25_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 133C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD25_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 133E                 | TCD Control and Status (DMA_TCD25_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 1340                 | TCD Source Address (DMA_TCD26_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 1344                 | TCD Transfer Attributes (DMA_TCD26_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 1346                 | TCD Signed Source Address Offset (DMA_TCD26_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 1348                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD26_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 1348                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD26_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 1348                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD26_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |

Table continues on the next page...

## DMA memory map (continued)

| Address offset (hex) | Register name                                                                                            | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|----------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 134C                 | TCD Last Source Address Adjustment (DMA_TCD26_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 1350                 | TCD Destination Address (DMA_TCD26_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 1354                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD26_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 1354                 | DMA_TCD26_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 1356                 | TCD Signed Destination Address Offset (DMA_TCD26_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 1358                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD26_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 135C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD26_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 135C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD26_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 135E                 | TCD Control and Status (DMA_TCD26_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 1360                 | TCD Source Address (DMA_TCD27_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 1364                 | TCD Transfer Attributes (DMA_TCD27_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 1366                 | TCD Signed Source Address Offset (DMA_TCD27_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 1368                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD27_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 1368                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD27_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 1368                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD27_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 136C                 | TCD Last Source Address Adjustment (DMA_TCD27_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 1370                 | TCD Destination Address (DMA_TCD27_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 1374                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD27_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 1374                 | DMA_TCD27_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 1376                 | TCD Signed Destination Address Offset (DMA_TCD27_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 1378                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD27_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |

Table continues on the next page...



**DMA memory map (continued)**

| Address offset (hex) | Register name                                                                                            | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|----------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 137C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD27_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 137C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD27_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 137E                 | TCD Control and Status (DMA_TCD27_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 1380                 | TCD Source Address (DMA_TCD28_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 1384                 | TCD Transfer Attributes (DMA_TCD28_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 1386                 | TCD Signed Source Address Offset (DMA_TCD28_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 1388                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD28_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 1388                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD28_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 1388                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD28_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 138C                 | TCD Last Source Address Adjustment (DMA_TCD28_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 1390                 | TCD Destination Address (DMA_TCD28_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 1394                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD28_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 1394                 | DMA_TCD28_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 1396                 | TCD Signed Destination Address Offset (DMA_TCD28_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 1398                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD28_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 139C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD28_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 139C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD28_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 139E                 | TCD Control and Status (DMA_TCD28_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 13A0                 | TCD Source Address (DMA_TCD29_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 13A4                 | TCD Transfer Attributes (DMA_TCD29_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |

*Table continues on the next page...*

## DMA memory map (continued)

| Address offset (hex) | Register name                                                                                            | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|----------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 13A6                 | TCD Signed Source Address Offset (DMA_TCD29_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 13A8                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD29_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 13A8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD29_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 13A8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD29_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 13AC                 | TCD Last Source Address Adjustment (DMA_TCD29_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 13B0                 | TCD Destination Address (DMA_TCD29_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 13B4                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD29_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 13B4                 | DMA_TCD29_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 13B6                 | TCD Signed Destination Address Offset (DMA_TCD29_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 13B8                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD29_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 13BC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD29_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 13BC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD29_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 13BE                 | TCD Control and Status (DMA_TCD29_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 13C0                 | TCD Source Address (DMA_TCD30_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 13C4                 | TCD Transfer Attributes (DMA_TCD30_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 13C6                 | TCD Signed Source Address Offset (DMA_TCD30_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 13C8                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD30_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 13C8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD30_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 13C8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD30_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 13CC                 | TCD Last Source Address Adjustment (DMA_TCD30_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 13D0                 | TCD Destination Address (DMA_TCD30_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |

Table continues on the next page...

## DMA memory map (continued)

| Address offset (hex) | Register name                                                                                            | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|----------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 13D4                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD30_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 13D4                 | DMA_TCD30_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 13D6                 | TCD Signed Destination Address Offset (DMA_TCD30_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 13D8                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD30_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 13DC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD30_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 13DC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD30_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 13DE                 | TCD Control and Status (DMA_TCD30_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 13E0                 | TCD Source Address (DMA_TCD31_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 13E4                 | TCD Transfer Attributes (DMA_TCD31_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 13E6                 | TCD Signed Source Address Offset (DMA_TCD31_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 13E8                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD31_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 13E8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD31_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 13E8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD31_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 13EC                 | TCD Last Source Address Adjustment (DMA_TCD31_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 13F0                 | TCD Destination Address (DMA_TCD31_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 13F4                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD31_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 13F4                 | DMA_TCD31_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 13F6                 | TCD Signed Destination Address Offset (DMA_TCD31_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 13F8                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD31_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 13FC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD31_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |

Table continues on the next page...

## DMA memory map (continued)

| Address offset (hex) | Register name                                                                                            | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|----------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 13FC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD31_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 13FE                 | TCD Control and Status (DMA_TCD31_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 1400                 | TCD Source Address (DMA_TCD32_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 1404                 | TCD Transfer Attributes (DMA_TCD32_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 1406                 | TCD Signed Source Address Offset (DMA_TCD32_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 1408                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD32_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 1408                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD32_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 1408                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD32_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 140C                 | TCD Last Source Address Adjustment (DMA_TCD32_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 1410                 | TCD Destination Address (DMA_TCD32_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 1414                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD32_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 1414                 | DMA_TCD32_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 1416                 | TCD Signed Destination Address Offset (DMA_TCD32_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 1418                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD32_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 141C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD32_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 141C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD32_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 141E                 | TCD Control and Status (DMA_TCD32_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 1420                 | TCD Source Address (DMA_TCD33_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 1424                 | TCD Transfer Attributes (DMA_TCD33_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 1426                 | TCD Signed Source Address Offset (DMA_TCD33_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 1428                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD33_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |

Table continues on the next page...

## DMA memory map (continued)

| Address offset (hex) | Register name                                                                                            | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|----------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 1428                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD33_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 1428                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD33_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 142C                 | TCD Last Source Address Adjustment (DMA_TCD33_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 1430                 | TCD Destination Address (DMA_TCD33_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 1434                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD33_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 1434                 | DMA_TCD33_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 1436                 | TCD Signed Destination Address Offset (DMA_TCD33_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 1438                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD33_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 143C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD33_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 143C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD33_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 143E                 | TCD Control and Status (DMA_TCD33_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 1440                 | TCD Source Address (DMA_TCD34_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 1444                 | TCD Transfer Attributes (DMA_TCD34_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 1446                 | TCD Signed Source Address Offset (DMA_TCD34_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 1448                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD34_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 1448                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD34_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 1448                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD34_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 144C                 | TCD Last Source Address Adjustment (DMA_TCD34_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 1450                 | TCD Destination Address (DMA_TCD34_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 1454                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD34_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 1454                 | DMA_TCD34_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |

Table continues on the next page...

## DMA memory map (continued)

| Address offset (hex) | Register name                                                                                            | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|----------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 1456                 | TCD Signed Destination Address Offset (DMA_TCD34_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 1458                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD34_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 145C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD34_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 145C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD34_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 145E                 | TCD Control and Status (DMA_TCD34_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 1460                 | TCD Source Address (DMA_TCD35_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 1464                 | TCD Transfer Attributes (DMA_TCD35_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 1466                 | TCD Signed Source Address Offset (DMA_TCD35_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 1468                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD35_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 1468                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD35_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 1468                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD35_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 146C                 | TCD Last Source Address Adjustment (DMA_TCD35_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 1470                 | TCD Destination Address (DMA_TCD35_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 1474                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD35_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 1474                 | DMA_TCD35_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 1476                 | TCD Signed Destination Address Offset (DMA_TCD35_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 1478                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD35_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 147C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD35_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 147C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD35_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 147E                 | TCD Control and Status (DMA_TCD35_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |

Table continues on the next page...



**DMA memory map (continued)**

| Address offset (hex) | Register name                                                                                            | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|----------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 1480                 | TCD Source Address (DMA_TCD36_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 1484                 | TCD Transfer Attributes (DMA_TCD36_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 1486                 | TCD Signed Source Address Offset (DMA_TCD36_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 1488                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD36_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 1488                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD36_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 1488                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD36_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 148C                 | TCD Last Source Address Adjustment (DMA_TCD36_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 1490                 | TCD Destination Address (DMA_TCD36_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 1494                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD36_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 1494                 | DMA_TCD36_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 1496                 | TCD Signed Destination Address Offset (DMA_TCD36_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 1498                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD36_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 149C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD36_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 149C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD36_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 149E                 | TCD Control and Status (DMA_TCD36_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 14A0                 | TCD Source Address (DMA_TCD37_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 14A4                 | TCD Transfer Attributes (DMA_TCD37_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 14A6                 | TCD Signed Source Address Offset (DMA_TCD37_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 14A8                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD37_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 14A8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD37_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 14A8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD37_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |

Table continues on the next page...

## DMA memory map (continued)

| Address offset (hex) | Register name                                                                                            | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|----------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 14AC                 | TCD Last Source Address Adjustment (DMA_TCD37_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 14B0                 | TCD Destination Address (DMA_TCD37_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 14B4                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD37_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 14B4                 | DMA_TCD37_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 14B6                 | TCD Signed Destination Address Offset (DMA_TCD37_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 14B8                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD37_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 14BC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD37_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 14BC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD37_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 14BE                 | TCD Control and Status (DMA_TCD37_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 14C0                 | TCD Source Address (DMA_TCD38_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 14C4                 | TCD Transfer Attributes (DMA_TCD38_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 14C6                 | TCD Signed Source Address Offset (DMA_TCD38_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 14C8                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD38_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 14C8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD38_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 14C8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD38_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 14CC                 | TCD Last Source Address Adjustment (DMA_TCD38_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 14D0                 | TCD Destination Address (DMA_TCD38_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 14D4                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD38_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 14D4                 | DMA_TCD38_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 14D6                 | TCD Signed Destination Address Offset (DMA_TCD38_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 14D8                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD38_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |

Table continues on the next page...



**DMA memory map (continued)**

| Address offset (hex) | Register name                                                                                            | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|----------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 14DC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD38_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 14DC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD38_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 14DE                 | TCD Control and Status (DMA_TCD38_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 14E0                 | TCD Source Address (DMA_TCD39_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 14E4                 | TCD Transfer Attributes (DMA_TCD39_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 14E6                 | TCD Signed Source Address Offset (DMA_TCD39_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 14E8                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD39_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 14E8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD39_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 14E8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD39_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 14EC                 | TCD Last Source Address Adjustment (DMA_TCD39_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 14F0                 | TCD Destination Address (DMA_TCD39_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 14F4                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD39_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 14F4                 | DMA_TCD39_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 14F6                 | TCD Signed Destination Address Offset (DMA_TCD39_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 14F8                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD39_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 14FC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD39_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 14FC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD39_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 14FE                 | TCD Control and Status (DMA_TCD39_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 1500                 | TCD Source Address (DMA_TCD40_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 1504                 | TCD Transfer Attributes (DMA_TCD40_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |

*Table continues on the next page...*

## DMA memory map (continued)

| Address offset (hex) | Register name                                                                                            | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|----------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 1506                 | TCD Signed Source Address Offset (DMA_TCD40_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 1508                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD40_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 1508                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD40_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 1508                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD40_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 150C                 | TCD Last Source Address Adjustment (DMA_TCD40_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 1510                 | TCD Destination Address (DMA_TCD40_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 1514                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD40_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 1514                 | DMA_TCD40_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 1516                 | TCD Signed Destination Address Offset (DMA_TCD40_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 1518                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD40_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 151C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD40_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 151C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD40_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 151E                 | TCD Control and Status (DMA_TCD40_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 1520                 | TCD Source Address (DMA_TCD41_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 1524                 | TCD Transfer Attributes (DMA_TCD41_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 1526                 | TCD Signed Source Address Offset (DMA_TCD41_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 1528                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD41_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 1528                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD41_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 1528                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD41_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 152C                 | TCD Last Source Address Adjustment (DMA_TCD41_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 1530                 | TCD Destination Address (DMA_TCD41_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |

Table continues on the next page...

**DMA memory map (continued)**

| Address offset (hex) | Register name                                                                                            | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|----------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 1534                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD41_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 1534                 | DMA_TCD41_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 1536                 | TCD Signed Destination Address Offset (DMA_TCD41_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 1538                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD41_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 153C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD41_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 153C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD41_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 153E                 | TCD Control and Status (DMA_TCD41_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 1540                 | TCD Source Address (DMA_TCD42_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 1544                 | TCD Transfer Attributes (DMA_TCD42_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 1546                 | TCD Signed Source Address Offset (DMA_TCD42_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 1548                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD42_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 1548                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD42_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 1548                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD42_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 154C                 | TCD Last Source Address Adjustment (DMA_TCD42_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 1550                 | TCD Destination Address (DMA_TCD42_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 1554                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD42_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 1554                 | DMA_TCD42_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 1556                 | TCD Signed Destination Address Offset (DMA_TCD42_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 1558                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD42_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 155C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD42_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |

*Table continues on the next page...*

## DMA memory map (continued)

| Address offset (hex) | Register name                                                                                            | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|----------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 155C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD42_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 155E                 | TCD Control and Status (DMA_TCD42_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 1560                 | TCD Source Address (DMA_TCD43_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 1564                 | TCD Transfer Attributes (DMA_TCD43_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 1566                 | TCD Signed Source Address Offset (DMA_TCD43_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 1568                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD43_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 1568                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD43_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 1568                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD43_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 156C                 | TCD Last Source Address Adjustment (DMA_TCD43_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 1570                 | TCD Destination Address (DMA_TCD43_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 1574                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD43_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 1574                 | DMA_TCD43_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 1576                 | TCD Signed Destination Address Offset (DMA_TCD43_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 1578                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD43_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 157C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD43_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 157C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD43_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 157E                 | TCD Control and Status (DMA_TCD43_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 1580                 | TCD Source Address (DMA_TCD44_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 1584                 | TCD Transfer Attributes (DMA_TCD44_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 1586                 | TCD Signed Source Address Offset (DMA_TCD44_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 1588                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD44_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |

Table continues on the next page...

**DMA memory map (continued)**

| Address offset (hex) | Register name                                                                                            | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|----------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 1588                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD44_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 1588                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD44_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 158C                 | TCD Last Source Address Adjustment (DMA_TCD44_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 1590                 | TCD Destination Address (DMA_TCD44_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 1594                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD44_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 1594                 | DMA_TCD44_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 1596                 | TCD Signed Destination Address Offset (DMA_TCD44_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 1598                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD44_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 159C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD44_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 159C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD44_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 159E                 | TCD Control and Status (DMA_TCD44_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 15A0                 | TCD Source Address (DMA_TCD45_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 15A4                 | TCD Transfer Attributes (DMA_TCD45_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 15A6                 | TCD Signed Source Address Offset (DMA_TCD45_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 15A8                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD45_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 15A8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD45_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 15A8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD45_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 15AC                 | TCD Last Source Address Adjustment (DMA_TCD45_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 15B0                 | TCD Destination Address (DMA_TCD45_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 15B4                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD45_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 15B4                 | DMA_TCD45_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |

*Table continues on the next page...*

## DMA memory map (continued)

| Address offset (hex) | Register name                                                                                            | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|----------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 15B6                 | TCD Signed Destination Address Offset (DMA_TCD45_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 15B8                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD45_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 15BC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD45_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 15BC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD45_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 15BE                 | TCD Control and Status (DMA_TCD45_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 15C0                 | TCD Source Address (DMA_TCD46_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 15C4                 | TCD Transfer Attributes (DMA_TCD46_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 15C6                 | TCD Signed Source Address Offset (DMA_TCD46_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 15C8                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD46_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 15C8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD46_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 15C8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD46_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 15CC                 | TCD Last Source Address Adjustment (DMA_TCD46_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 15D0                 | TCD Destination Address (DMA_TCD46_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 15D4                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD46_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 15D4                 | DMA_TCD46_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 15D6                 | TCD Signed Destination Address Offset (DMA_TCD46_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 15D8                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD46_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 15DC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD46_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 15DC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD46_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 15DE                 | TCD Control and Status (DMA_TCD46_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |

Table continues on the next page...



**DMA memory map (continued)**

| Address offset (hex) | Register name                                                                                            | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|----------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 15E0                 | TCD Source Address (DMA_TCD47_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 15E4                 | TCD Transfer Attributes (DMA_TCD47_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 15E6                 | TCD Signed Source Address Offset (DMA_TCD47_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 15E8                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD47_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 15E8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD47_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 15E8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD47_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 15EC                 | TCD Last Source Address Adjustment (DMA_TCD47_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 15F0                 | TCD Destination Address (DMA_TCD47_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 15F4                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD47_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 15F4                 | DMA_TCD47_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 15F6                 | TCD Signed Destination Address Offset (DMA_TCD47_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 15F8                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD47_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 15FC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD47_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 15FC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD47_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 15FE                 | TCD Control and Status (DMA_TCD47_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 1600                 | TCD Source Address (DMA_TCD48_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 1604                 | TCD Transfer Attributes (DMA_TCD48_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 1606                 | TCD Signed Source Address Offset (DMA_TCD48_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 1608                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD48_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 1608                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD48_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 1608                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD48_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |

*Table continues on the next page...*

## DMA memory map (continued)

| Address offset (hex) | Register name                                                                                            | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|----------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 160C                 | TCD Last Source Address Adjustment (DMA_TCD48_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 1610                 | TCD Destination Address (DMA_TCD48_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 1614                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD48_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 1614                 | DMA_TCD48_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 1616                 | TCD Signed Destination Address Offset (DMA_TCD48_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 1618                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD48_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 161C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD48_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 161C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD48_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 161E                 | TCD Control and Status (DMA_TCD48_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 1620                 | TCD Source Address (DMA_TCD49_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 1624                 | TCD Transfer Attributes (DMA_TCD49_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 1626                 | TCD Signed Source Address Offset (DMA_TCD49_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 1628                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD49_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 1628                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD49_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 1628                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD49_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 162C                 | TCD Last Source Address Adjustment (DMA_TCD49_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 1630                 | TCD Destination Address (DMA_TCD49_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 1634                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD49_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 1634                 | DMA_TCD49_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 1636                 | TCD Signed Destination Address Offset (DMA_TCD49_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 1638                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD49_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |

Table continues on the next page...



**DMA memory map (continued)**

| Address offset (hex) | Register name                                                                                            | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|----------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 163C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD49_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 163C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD49_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 163E                 | TCD Control and Status (DMA_TCD49_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 1640                 | TCD Source Address (DMA_TCD50_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 1644                 | TCD Transfer Attributes (DMA_TCD50_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 1646                 | TCD Signed Source Address Offset (DMA_TCD50_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 1648                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD50_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 1648                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD50_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 1648                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD50_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 164C                 | TCD Last Source Address Adjustment (DMA_TCD50_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 1650                 | TCD Destination Address (DMA_TCD50_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 1654                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD50_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 1654                 | DMA_TCD50_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 1656                 | TCD Signed Destination Address Offset (DMA_TCD50_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 1658                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD50_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 165C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD50_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 165C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD50_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 165E                 | TCD Control and Status (DMA_TCD50_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 1660                 | TCD Source Address (DMA_TCD51_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 1664                 | TCD Transfer Attributes (DMA_TCD51_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |

*Table continues on the next page...*

## DMA memory map (continued)

| Address offset (hex) | Register name                                                                                            | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|----------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 1666                 | TCD Signed Source Address Offset (DMA_TCD51_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 1668                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD51_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 1668                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD51_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 1668                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD51_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 166C                 | TCD Last Source Address Adjustment (DMA_TCD51_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 1670                 | TCD Destination Address (DMA_TCD51_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 1674                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD51_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 1674                 | DMA_TCD51_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 1676                 | TCD Signed Destination Address Offset (DMA_TCD51_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 1678                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD51_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 167C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD51_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 167C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD51_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 167E                 | TCD Control and Status (DMA_TCD51_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 1680                 | TCD Source Address (DMA_TCD52_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 1684                 | TCD Transfer Attributes (DMA_TCD52_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 1686                 | TCD Signed Source Address Offset (DMA_TCD52_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 1688                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD52_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 1688                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD52_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 1688                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD52_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 168C                 | TCD Last Source Address Adjustment (DMA_TCD52_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 1690                 | TCD Destination Address (DMA_TCD52_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |

Table continues on the next page...

**DMA memory map (continued)**

| Address offset (hex) | Register name                                                                                            | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|----------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 1694                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD52_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 1694                 | DMA_TCD52_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 1696                 | TCD Signed Destination Address Offset (DMA_TCD52_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 1698                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD52_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 169C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD52_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 169C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD52_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 169E                 | TCD Control and Status (DMA_TCD52_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 16A0                 | TCD Source Address (DMA_TCD53_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 16A4                 | TCD Transfer Attributes (DMA_TCD53_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 16A6                 | TCD Signed Source Address Offset (DMA_TCD53_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 16A8                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD53_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 16A8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD53_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 16A8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD53_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 16AC                 | TCD Last Source Address Adjustment (DMA_TCD53_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 16B0                 | TCD Destination Address (DMA_TCD53_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 16B4                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD53_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 16B4                 | DMA_TCD53_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 16B6                 | TCD Signed Destination Address Offset (DMA_TCD53_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 16B8                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD53_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 16BC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD53_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |

*Table continues on the next page...*

## DMA memory map (continued)

| Address offset (hex) | Register name                                                                                            | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|----------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 16BC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD53_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 16BE                 | TCD Control and Status (DMA_TCD53_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 16C0                 | TCD Source Address (DMA_TCD54_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 16C4                 | TCD Transfer Attributes (DMA_TCD54_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 16C6                 | TCD Signed Source Address Offset (DMA_TCD54_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 16C8                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD54_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 16C8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD54_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 16C8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD54_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 16CC                 | TCD Last Source Address Adjustment (DMA_TCD54_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 16D0                 | TCD Destination Address (DMA_TCD54_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 16D4                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD54_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 16D4                 | DMA_TCD54_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 16D6                 | TCD Signed Destination Address Offset (DMA_TCD54_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 16D8                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD54_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 16DC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD54_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 16DC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD54_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 16DE                 | TCD Control and Status (DMA_TCD54_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 16E0                 | TCD Source Address (DMA_TCD55_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 16E4                 | TCD Transfer Attributes (DMA_TCD55_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 16E6                 | TCD Signed Source Address Offset (DMA_TCD55_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 16E8                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD55_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |

Table continues on the next page...

**DMA memory map (continued)**

| Address offset (hex) | Register name                                                                                            | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|----------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 16E8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD55_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 16E8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD55_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 16EC                 | TCD Last Source Address Adjustment (DMA_TCD55_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 16F0                 | TCD Destination Address (DMA_TCD55_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 16F4                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD55_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 16F4                 | DMA_TCD55_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 16F6                 | TCD Signed Destination Address Offset (DMA_TCD55_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 16F8                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD55_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 16FC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD55_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 16FC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD55_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 16FE                 | TCD Control and Status (DMA_TCD55_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 1700                 | TCD Source Address (DMA_TCD56_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 1704                 | TCD Transfer Attributes (DMA_TCD56_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 1706                 | TCD Signed Source Address Offset (DMA_TCD56_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 1708                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD56_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 1708                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD56_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 1708                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD56_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 170C                 | TCD Last Source Address Adjustment (DMA_TCD56_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 1710                 | TCD Destination Address (DMA_TCD56_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 1714                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD56_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 1714                 | DMA_TCD56_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |

*Table continues on the next page...*

## DMA memory map (continued)

| Address offset (hex) | Register name                                                                                            | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|----------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 1716                 | TCD Signed Destination Address Offset (DMA_TCD56_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 1718                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD56_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 171C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD56_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 171C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD56_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 171E                 | TCD Control and Status (DMA_TCD56_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 1720                 | TCD Source Address (DMA_TCD57_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 1724                 | TCD Transfer Attributes (DMA_TCD57_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 1726                 | TCD Signed Source Address Offset (DMA_TCD57_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 1728                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD57_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 1728                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD57_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 1728                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD57_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 172C                 | TCD Last Source Address Adjustment (DMA_TCD57_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 1730                 | TCD Destination Address (DMA_TCD57_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 1734                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD57_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 1734                 | DMA_TCD57_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 1736                 | TCD Signed Destination Address Offset (DMA_TCD57_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 1738                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD57_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 173C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD57_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 173C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD57_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 173E                 | TCD Control and Status (DMA_TCD57_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |

Table continues on the next page...



**DMA memory map (continued)**

| Address offset (hex) | Register name                                                                                            | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|----------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 1740                 | TCD Source Address (DMA_TCD58_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 1744                 | TCD Transfer Attributes (DMA_TCD58_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 1746                 | TCD Signed Source Address Offset (DMA_TCD58_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 1748                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD58_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 1748                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD58_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 1748                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD58_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 174C                 | TCD Last Source Address Adjustment (DMA_TCD58_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 1750                 | TCD Destination Address (DMA_TCD58_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 1754                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD58_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 1754                 | DMA_TCD58_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 1756                 | TCD Signed Destination Address Offset (DMA_TCD58_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 1758                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD58_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 175C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD58_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 175C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD58_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 175E                 | TCD Control and Status (DMA_TCD58_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 1760                 | TCD Source Address (DMA_TCD59_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 1764                 | TCD Transfer Attributes (DMA_TCD59_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 1766                 | TCD Signed Source Address Offset (DMA_TCD59_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 1768                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD59_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 1768                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD59_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 1768                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD59_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |

Table continues on the next page...

## DMA memory map (continued)

| Address offset (hex) | Register name                                                                                            | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|----------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 176C                 | TCD Last Source Address Adjustment (DMA_TCD59_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 1770                 | TCD Destination Address (DMA_TCD59_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 1774                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD59_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 1774                 | DMA_TCD59_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 1776                 | TCD Signed Destination Address Offset (DMA_TCD59_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 1778                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD59_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 177C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD59_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 177C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD59_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 177E                 | TCD Control and Status (DMA_TCD59_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 1780                 | TCD Source Address (DMA_TCD60_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 1784                 | TCD Transfer Attributes (DMA_TCD60_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 1786                 | TCD Signed Source Address Offset (DMA_TCD60_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 1788                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD60_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 1788                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD60_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 1788                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD60_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 178C                 | TCD Last Source Address Adjustment (DMA_TCD60_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 1790                 | TCD Destination Address (DMA_TCD60_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 1794                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD60_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 1794                 | DMA_TCD60_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 1796                 | TCD Signed Destination Address Offset (DMA_TCD60_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 1798                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD60_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |

Table continues on the next page...



## DMA memory map (continued)

| Address offset (hex) | Register name                                                                                            | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|----------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 179C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD60_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 179C                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD60_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 179E                 | TCD Control and Status (DMA_TCD60_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 17A0                 | TCD Source Address (DMA_TCD61_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 17A4                 | TCD Transfer Attributes (DMA_TCD61_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 17A6                 | TCD Signed Source Address Offset (DMA_TCD61_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 17A8                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD61_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 17A8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD61_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 17A8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD61_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 17AC                 | TCD Last Source Address Adjustment (DMA_TCD61_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 17B0                 | TCD Destination Address (DMA_TCD61_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 17B4                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD61_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 17B4                 | DMA_TCD61_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 17B6                 | TCD Signed Destination Address Offset (DMA_TCD61_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 17B8                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD61_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 17BC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD61_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 17BC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD61_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 17BE                 | TCD Control and Status (DMA_TCD61_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 17C0                 | TCD Source Address (DMA_TCD62_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 17C4                 | TCD Transfer Attributes (DMA_TCD62_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |

Table continues on the next page...

## DMA memory map (continued)

| Address offset (hex) | Register name                                                                                            | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|----------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 17C6                 | TCD Signed Source Address Offset (DMA_TCD62_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 17C8                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD62_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 17C8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD62_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 17C8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD62_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 17CC                 | TCD Last Source Address Adjustment (DMA_TCD62_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 17D0                 | TCD Destination Address (DMA_TCD62_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |
| 17D4                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD62_CITER_ELINKYES)       | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 17D4                 | DMA_TCD62_CITER_ELINKNO                                                                                  | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 17D6                 | TCD Signed Destination Address Offset (DMA_TCD62_DOFF)                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 17D8                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD62_DLASTSGA)                      | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 17DC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD62_BITER_ELINKYES)     | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 17DC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD62_BITER_ELINKNO)     | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 17DE                 | TCD Control and Status (DMA_TCD62_CSR)                                                                   | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |
| 17E0                 | TCD Source Address (DMA_TCD63_SADDR)                                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.27/ 674</a> |
| 17E4                 | TCD Transfer Attributes (DMA_TCD63_ATTR)                                                                 | 16              | R/W    | 0000h       | <a href="#">16.4.28/ 674</a> |
| 17E6                 | TCD Signed Source Address Offset (DMA_TCD63_SOFF)                                                        | 16              | R/W    | 0000h       | <a href="#">16.4.29/ 675</a> |
| 17E8                 | TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA_TCD63_NBYTES_MLNO)                               | 32              | R/W    | 0000_0000h  | <a href="#">16.4.30/ 675</a> |
| 17E8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA_TCD63_NBYTES_MLOFFNO) | 32              | R/W    | 0000_0000h  | <a href="#">16.4.31/ 676</a> |
| 17E8                 | TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA_TCD63_NBYTES_MLOFFYES)         | 32              | R/W    | 0000_0000h  | <a href="#">16.4.32/ 677</a> |
| 17EC                 | TCD Last Source Address Adjustment (DMA_TCD63_SLAST)                                                     | 32              | R/W    | 0000_0000h  | <a href="#">16.4.33/ 678</a> |
| 17F0                 | TCD Destination Address (DMA_TCD63_DADDR)                                                                | 32              | R/W    | 0000_0000h  | <a href="#">16.4.34/ 679</a> |

Table continues on the next page...

**DMA memory map (continued)**

| Address offset (hex) | Register name                                                                                        | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|------------------------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 17F4                 | TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD63_CITER_ELINKYES)   | 16              | R/W    | 0000h       | <a href="#">16.4.35/ 679</a> |
| 17F4                 | DMA_TCD63_CITER_ELINKNO                                                                              | 16              | R/W    | 0000h       | <a href="#">16.4.36/ 680</a> |
| 17F6                 | TCD Signed Destination Address Offset (DMA_TCD63_DOFF)                                               | 16              | R/W    | 0000h       | <a href="#">16.4.37/ 681</a> |
| 17F8                 | TCD Last Destination Address Adjustment/Scatter Gather Address (DMA_TCD63_DLASTSGA)                  | 32              | R/W    | 0000_0000h  | <a href="#">16.4.38/ 682</a> |
| 17FC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA_TCD63_BITER_ELINKYES) | 16              | R/W    | 0000h       | <a href="#">16.4.39/ 682</a> |
| 17FC                 | TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA_TCD63_BITER_ELINKNO) | 16              | R/W    | 0000h       | <a href="#">16.4.40/ 683</a> |
| 17FE                 | TCD Control and Status (DMA_TCD63_CSR)                                                               | 16              | R/W    | 0000h       | <a href="#">16.4.41/ 684</a> |

**16.4.5 Control Register (DMA\_CR)**

The CR defines the basic operating configuration of the DMA. The DMA arbitrates channel service requests in four groups of 16 channels each:

- Group 3 contains channels 63-48
- Group 2 contains channels 47-32
- Group 1 contains channels 31-16
- Group 0 contains channels 15-0

Arbitration within a group can be configured to use either a fixed-priority or a round-robin scheme. For fixed-priority arbitration, the highest priority channel requesting service is selected to execute. The channel priority registers assign the priorities; see the DCHPRIn registers. For round-robin arbitration, the channel priorities are ignored and channels within each group are cycled through (from high to low channel number) without regard to priority.

**NOTE**

For correct operation, writes to the CR register must be performed only when the DMA channels are inactive; that is, when TCDn\_CSR[ACTIVE] bits are cleared.

The group priorities operate in a similar fashion. In group fixed priority arbitration mode, channel service requests in the highest priority group are executed first, where priority level 3 is the highest and priority level 0 is the lowest. The group priorities are assigned

in the GRPnPRI fields of the DMA Control Register (CR). All group priorities must have unique values prior to any channel service requests occurring; otherwise, a configuration error will be reported. For group round robin arbitration, the group priorities are ignored and the groups are cycled through (from high to low group number) without regard to priority.

Minor loop offsets are address offset values added to the final source address (TCDn\_SADDR) or destination address (TCDn\_DADDR) upon minor loop completion. When minor loop offsets are enabled, the minor loop offset (MLOFF) is added to the final source address (TCDn\_SADDR), to the final destination address (TCDn\_DADDR), or to both prior to the addresses being written back into the TCD. If the major loop is complete, the minor loop offset is ignored and the major loop address offsets (TCDn\_SLAST and TCDn\_DLAST\_SGA) are used to compute the next TCDn\_SADDR and TCDn\_DADDR values.

When minor loop mapping is enabled (EMLM is 1), TCDn word2 is redefined. A portion of TCDn word2 is used to specify multiple fields: a source enable bit (SMLOE) to specify the minor loop offset should be applied to the source address (TCDn\_SADDR) upon minor loop completion, a destination enable bit (DMLOE) to specify the minor loop offset should be applied to the destination address (TCDn\_DADDR) upon minor loop completion, and the sign extended minor loop offset value (MLOFF). The same offset value (MLOFF) is used for both source and destination minor loop offsets. When either minor loop offset is enabled (SMLOE set or DMLOE set), the NBYTES field is reduced to 10 bits. When both minor loop offsets are disabled (SMLOE cleared and DMLOE cleared), the NBYTES field is a 30-bit vector.

When minor loop mapping is disabled (EMLM is 0), all 32 bits of TCDn word2 are assigned to the NBYTES field.

Address: 0h base + 0h offset = 0h

| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14  | 15 |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|-----|----|
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    | CX | ECX |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |     |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0   | 0  |

| Bit   | 16      | 17 | 18      | 19 | 20      | 21 | 22      | 23 | 24   | 25  | 26   | 27  | 28   | 29   | 30   | 31       |
|-------|---------|----|---------|----|---------|----|---------|----|------|-----|------|-----|------|------|------|----------|
| R     | GRP3PRI |    | GRP2PRI |    | GRP1PRI |    | GRP0PRI |    | EMLM | CLM | HALT | HOE | ERGA | ERCA | EDBG | Reserved |
| W     |         |    |         |    |         |    |         |    |      |     |      |     |      |      |      |          |
| Reset | 1       | 1  | 1       | 0  | 0       | 1  | 0       | 0  | 0    | 0   | 0    | 0   | 0    | 0    | 0    | 0        |

**DMA\_CR field descriptions**

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–13<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 14<br>CX         | Cancel Transfer<br><br>0 Normal operation<br>1 Cancel the remaining data transfer. Stop the executing channel and force the minor loop to finish. The cancel takes effect after the last write of the current read/write sequence. The CX bit clears itself after the cancel has been honored. This cancel retires the channel normally as if the minor loop was completed.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 15<br>ECX        | Error Cancel Transfer<br><br>0 Normal operation<br>1 Cancel the remaining data transfer in the same fashion as the CX bit. Stop the executing channel and force the minor loop to finish. The cancel takes effect after the last write of the current read/write sequence. The ECX bit clears itself after the cancel is honored. In addition to cancelling the transfer, ECX treats the cancel as an error condition, thus updating the Error Status register (DMAx_ES) and generating an optional error interrupt.                                                                                                                                                                                                                                                                                                                                                            |
| 16–17<br>GRP3PRI | Channel Group 3 Priority<br><br>Group 3 priority level when fixed priority group arbitration is enabled.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 18–19<br>GRP2PRI | Channel Group 2 Priority<br><br>Group 2 priority level when fixed priority group arbitration is enabled.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 20–21<br>GRP1PRI | Channel Group 1 Priority<br><br>Group 1 priority level when fixed priority group arbitration is enabled.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 22–23<br>GRP0PRI | Channel Group 0 Priority<br><br>Group 0 priority level when fixed priority group arbitration is enabled.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 24<br>EMLM       | Enable Minor Loop Mapping<br><br>0 Disabled. TCDn.word2 is defined as a 32-bit NBYTES field.<br>1 Enabled. TCDn.word2 is redefined to include individual enable fields, an offset field, and the NBYTES field. The individual enable fields allow the minor loop offset to be applied to the source address, the destination address, or both. The NBYTES field is reduced when either offset is enabled.                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 25<br>CLM        | Continuous Link Mode<br><br><b>NOTE:</b> Do not use continuous link mode with a channel linking to itself if there is only one minor loop iteration per service request, e.g., if the channel's NBYTES value is the same as either the source or destination size. The same data transfer profile can be achieved by simply increasing the NBYTES value, which provides more efficient, faster processing.<br><br>0 A minor loop channel link made to itself goes through channel arbitration before being activated again.<br>1 A minor loop channel link made to itself does not go through channel arbitration before being activated again. Upon minor loop completion, the channel activates again if that channel has a minor loop channel link enabled and the link channel is itself. This effectively applies the minor loop offsets and restarts the next minor loop. |
| 26<br>HALT       | Halt DMA Operations                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

*Table continues on the next page...*

**DMA\_CR field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                                |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | 0 Normal operation<br>1 Stall the start of any new channels. Executing channels are allowed to complete. Channel execution resumes when this bit is cleared.                                                                                                               |
| 27<br>HOE      | Halt On Error<br>0 Normal operation<br>1 Any error causes the HALT bit to set. Subsequently, all service requests are ignored until the HALT bit is cleared.                                                                                                               |
| 28<br>ERGA     | Enable Round Robin Group Arbitration<br>0 Fixed priority arbitration is used for selection among the groups.<br>1 Round robin arbitration is used for selection among the groups.                                                                                          |
| 29<br>ERCA     | Enable Round Robin Channel Arbitration<br>0 Fixed priority arbitration is used for channel selection within each group.<br>1 Round robin arbitration is used for channel selection within each group.                                                                      |
| 30<br>EDBG     | Enable Debug<br>0 When in debug mode, the DMA continues to operate.<br>1 When in debug mode, the DMA stalls the start of a new channel. Executing channels are allowed to complete. Channel execution resumes when the system exits debug mode or the EDBG bit is cleared. |
| 31<br>Reserved | This field is reserved.<br>Reserved                                                                                                                                                                                                                                        |

**16.4.6 Error Status Register (DMA\_ES)**

The ES provides information concerning the last recorded channel error. Channel errors can be caused by:

- A configuration error, that is:
  - An illegal setting in the transfer-control descriptor, or
  - An illegal priority register setting in fixed-arbitration
- An error termination to a bus master read or write cycle
- The UCE bit shows an uncorrectable error occurred while the device has ECC protection on the TCD SRAM
- A cancel transfer with error bit that will be set when a transfer is canceled via the corresponding cancel transfer control bit

See [Fault reporting and handling](#) for more details.

Address: 0h base + 4h offset = 4h

|       |     |   |   |   |   |   |   |   |   |   |    |    |    |    |     |     |
|-------|-----|---|---|---|---|---|---|---|---|---|----|----|----|----|-----|-----|
| Bit   | 0   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14  | 15  |
| R     | VLD | 0 |   |   |   |   |   |   |   |   |    |    |    |    | UCE | ECX |
| W     |     |   |   |   |   |   |   |   |   |   |    |    |    |    |     |     |
| Reset | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0   | 0   |

|       |     |     |        |    |    |    |    |    |     |     |     |     |     |     |     |     |
|-------|-----|-----|--------|----|----|----|----|----|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit   | 16  | 17  | 18     | 19 | 20 | 21 | 22 | 23 | 24  | 25  | 26  | 27  | 28  | 29  | 30  | 31  |
| R     | GPE | CPE | ERRCHN |    |    |    |    |    | SAE | SOE | DAE | DOE | NCE | SGE | SBE | DBE |
| W     |     |     |        |    |    |    |    |    |     |     |     |     |     |     |     |     |
| Reset | 0   | 0   | 0      | 0  | 0  | 0  | 0  | 0  | 0   | 0   | 0   | 0   | 0   | 0   | 0   | 0   |

**DMA\_ES field descriptions**

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>VLD         | Logical OR of all ERRH and ERRL status bits<br><br>0 No ERR bits are set.<br>1 At least one ERR bit is set indicating a valid error exists that has not been cleared.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 1–13<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 14<br>UCE        | Uncorrectable ECC error during channel execution.<br><br>The UCE bit is set when an uncorrectable ECC error occurs on an access generated by the DMA only. If a CPU access to the TCD causes an uncorrectable ECC error, that access will receive a bus error response.<br><br><b>NOTE:</b> When the eDMA sees a RAM error on an IPS access (when the user is accessing a TCD), the error is reported as a bus abort. When the dma_engine (the execution engine is accessing a TCD) receives a RAM error, it is recorded in the Error Status register, DMA_ES[UCE], along with the channel number.<br><br>0 No uncorrectable ECC error<br>1 The last recorded error was an uncorrectable TCD RAM error |
| 15<br>ECX        | Transfer Canceled<br><br>0 No canceled transfers<br>1 The last recorded entry was a canceled transfer by the error cancel transfer input                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 16<br>GPE        | Group Priority Error<br><br>0 No group priority error<br>1 The last recorded error was a configuration error among the group priorities. All group priorities are not unique.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 17<br>CPE        | Channel Priority Error<br><br>0 No channel priority error<br>1 The last recorded error was a configuration error in the channel priorities within a group. Channel priorities within a group are not unique.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 18–23<br>ERRCHN  | Error Channel Number or Canceled Channel Number<br><br>The channel number of the last recorded error, excluding GPE and CPE errors, or last recorded error canceled transfer.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

*Table continues on the next page...*

**DMA\_ES field descriptions (continued)**

| Field     | Description                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 24<br>SAE | Source Address Error<br><br>0 No source address configuration error.<br>1 The last recorded error was a configuration error detected in the TCDn_SADDR field. TCDn_SADDR is inconsistent with TCDn_ATTR[SSIZE].                                                                                                                                                                                                                           |
| 25<br>SOE | Source Offset Error<br><br>0 No source offset configuration error<br>1 The last recorded error was a configuration error detected in the TCDn_SOFF field. TCDn_SOFF is inconsistent with TCDn_ATTR[SSIZE].                                                                                                                                                                                                                                |
| 26<br>DAE | Destination Address Error<br><br>0 No destination address configuration error<br>1 The last recorded error was a configuration error detected in the TCDn_DADDR field. TCDn_DADDR is inconsistent with TCDn_ATTR[DSIZE].                                                                                                                                                                                                                  |
| 27<br>DOE | Destination Offset Error<br><br>0 No destination offset configuration error<br>1 The last recorded error was a configuration error detected in the TCDn_DOFF field. TCDn_DOFF is inconsistent with TCDn_ATTR[DSIZE].                                                                                                                                                                                                                      |
| 28<br>NCE | NBYTES/CITER Configuration Error<br><br>0 No NBYTES/CITER configuration error<br>1 The last recorded error was a configuration error detected in the TCDn_NBYTES or TCDn_CITER fields.<br><ul style="list-style-type: none"> <li>• TCDn_NBYTES is not a multiple of TCDn_ATTR[SSIZE] and TCDn_ATTR[DSIZE], or</li> <li>• TCDn_CITER[CITER] is equal to zero, or</li> <li>• TCDn_CITER[ELINK] is not equal to TCDn_BITER[ELINK]</li> </ul> |
| 29<br>SGE | Scatter/Gather Configuration Error<br><br>0 No scatter/gather configuration error<br>1 The last recorded error was a configuration error detected in the TCDn_DLASTSGA field. This field is checked at the beginning of a scatter/gather operation after major loop completion if TCDn_CSR[ESG] is enabled. TCDn_DLASTSGA is not on a 32 byte boundary.                                                                                   |
| 30<br>SBE | Source Bus Error<br><br>0 No source bus error<br>1 The last recorded error was a bus error on a source read                                                                                                                                                                                                                                                                                                                               |
| 31<br>DBE | Destination Bus Error<br><br>0 No destination bus error<br>1 The last recorded error was a bus error on a destination write                                                                                                                                                                                                                                                                                                               |



### 16.4.7 Enable Request Register High (DMA\_ERQH)

The ERQ{H,L} registers provide a bit map for the 64 channels to enable the request signal for each channel. ERQH supports channels 63-32, while ERQL covers channels 31-00. The state of any given channel enable is directly affected by writes to this register; it is also affected by writes to the SERQ and CERQ. The {S,C}ERQ registers are provided so the request enable for a single channel can easily be modified without needing to perform a read-modify-write sequence to the ERQ{H,L}.

DMA request input signals and this enable request flag must be asserted before a channel's hardware service request is accepted. The state of the eDMA enable request flag does not affect a channel service request made explicitly through software or a linked channel request.

#### NOTE

Disable a channel's hardware service request at the source before clearing the channel's ERQ bit.

Address: 0h base + 8h offset = 8h

| Bit   | 0     | 1     | 2     | 3     | 4     | 5     | 6     | 7     | 8     | 9     | 10    | 11    | 12    | 13    | 14    | 15    |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| R     | ERQ63 | ERQ62 | ERQ61 | ERQ60 | ERQ59 | ERQ58 | ERQ57 | ERQ56 | ERQ55 | ERQ54 | ERQ53 | ERQ52 | ERQ51 | ERQ50 | ERQ49 | ERQ48 |
| W     |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| Bit   | 16    | 17    | 18    | 19    | 20    | 21    | 22    | 23    | 24    | 25    | 26    | 27    | 28    | 29    | 30    | 31    |
| R     | ERQ47 | ERQ46 | ERQ45 | ERQ44 | ERQ43 | ERQ42 | ERQ41 | ERQ40 | ERQ39 | ERQ38 | ERQ37 | ERQ36 | ERQ35 | ERQ34 | ERQ33 | ERQ32 |
| W     |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

#### DMA\_ERQH field descriptions

| Field      | Description                                                                                                                                                      |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>ERQ63 | Enable DMA Request 63<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 1<br>ERQ62 | Enable DMA Request 62<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |

Table continues on the next page...

**DMA\_ERQH field descriptions (continued)**

| Field       | Description                                                                                                                                                      |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2<br>ERQ61  | Enable DMA Request 61<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 3<br>ERQ60  | Enable DMA Request 60<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 4<br>ERQ59  | Enable DMA Request 59<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 5<br>ERQ58  | Enable DMA Request 58<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 6<br>ERQ57  | Enable DMA Request 57<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 7<br>ERQ56  | Enable DMA Request 56<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 8<br>ERQ55  | Enable DMA Request 55<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 9<br>ERQ54  | Enable DMA Request 54<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 10<br>ERQ53 | Enable DMA Request 53<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 11<br>ERQ52 | Enable DMA Request 52<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 12<br>ERQ51 | Enable DMA Request 51<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 13<br>ERQ50 | Enable DMA Request 50<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |

*Table continues on the next page...*

**DMA\_ERQH field descriptions (continued)**

| <b>Field</b> | <b>Description</b>                                                                                                                                               |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 14<br>ERQ49  | Enable DMA Request 49<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 15<br>ERQ48  | Enable DMA Request 48<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 16<br>ERQ47  | Enable DMA Request 47<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 17<br>ERQ46  | Enable DMA Request 46<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 18<br>ERQ45  | Enable DMA Request 45<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 19<br>ERQ44  | Enable DMA Request 44<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 20<br>ERQ43  | Enable DMA Request 43<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 21<br>ERQ42  | Enable DMA Request 42<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 22<br>ERQ41  | Enable DMA Request 41<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 23<br>ERQ40  | Enable DMA Request 40<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 24<br>ERQ39  | Enable DMA Request 39<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 25<br>ERQ38  | Enable DMA Request 38<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |

*Table continues on the next page...*

**DMA\_ERQH field descriptions (continued)**

| Field       | Description                                                                                                                                                      |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 26<br>ERQ37 | Enable DMA Request 37<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 27<br>ERQ36 | Enable DMA Request 36<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 28<br>ERQ35 | Enable DMA Request 35<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 29<br>ERQ34 | Enable DMA Request 34<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 30<br>ERQ33 | Enable DMA Request 33<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 31<br>ERQ32 | Enable DMA Request 32<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |

**16.4.8 Enable Request Register Low (DMA\_ERQL)**

The ERQ{H,L} registers provide a bit map for the 64 channels to enable the request signal for each channel. ERQH supports channels 63-32, while ERQL covers channels 31-00. The state of any given channel enable is directly affected by writes to this register; it is also affected by writes to the SERQ and CERQ registers. These registers are provided so the request enable for a single channel can easily be modified without needing to perform a read-modify-write sequence to the ERQ{H,L}.

DMA request input signals and this enable request flag must be asserted before a channel's hardware service request is accepted. The state of the DMA enable request flag does not affect a channel service request made explicitly through software or a linked channel request.

**NOTE**

Disable a channel's hardware service request at the source before clearing the channel's ERQ bit.

Address: 0h base + Ch offset = Ch

|       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit   | 0     | 1     | 2     | 3     | 4     | 5     | 6     | 7     | 8     | 9     | 10    | 11    | 12    | 13    | 14    | 15    |
| R     |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| W     | ERQ31 | ERQ30 | ERQ29 | ERQ28 | ERQ27 | ERQ26 | ERQ25 | ERQ24 | ERQ23 | ERQ22 | ERQ21 | ERQ20 | ERQ19 | ERQ18 | ERQ17 | ERQ16 |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

|       |       |       |       |       |       |       |      |      |      |      |      |      |      |      |      |      |
|-------|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| Bit   | 16    | 17    | 18    | 19    | 20    | 21    | 22   | 23   | 24   | 25   | 26   | 27   | 28   | 29   | 30   | 31   |
| R     |       |       |       |       |       |       |      |      |      |      |      |      |      |      |      |      |
| W     | ERQ15 | ERQ14 | ERQ13 | ERQ12 | ERQ11 | ERQ10 | ERQ9 | ERQ8 | ERQ7 | ERQ6 | ERQ5 | ERQ4 | ERQ3 | ERQ2 | ERQ1 | ERQ0 |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |

**DMA\_ERQL field descriptions**

| Field      | Description                                                                                                                                                      |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>ERQ31 | Enable DMA Request 31<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 1<br>ERQ30 | Enable DMA Request 30<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 2<br>ERQ29 | Enable DMA Request 29<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 3<br>ERQ28 | Enable DMA Request 28<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 4<br>ERQ27 | Enable DMA Request 27<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 5<br>ERQ26 | Enable DMA Request 26<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 6<br>ERQ25 | Enable DMA Request 25<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 7<br>ERQ24 | Enable DMA Request 24<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |

*Table continues on the next page...*

**DMA\_ERQL field descriptions (continued)**

| Field       | Description                                                                                                                                                      |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 8<br>ERQ23  | Enable DMA Request 23<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 9<br>ERQ22  | Enable DMA Request 22<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 10<br>ERQ21 | Enable DMA Request 21<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 11<br>ERQ20 | Enable DMA Request 20<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 12<br>ERQ19 | Enable DMA Request 19<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 13<br>ERQ18 | Enable DMA Request 18<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 14<br>ERQ17 | Enable DMA Request 17<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 15<br>ERQ16 | Enable DMA Request 16<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 16<br>ERQ15 | Enable DMA Request 15<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 17<br>ERQ14 | Enable DMA Request 14<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 18<br>ERQ13 | Enable DMA Request 13<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 19<br>ERQ12 | Enable DMA Request 12<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |

*Table continues on the next page...*

**DMA\_ERQL field descriptions (continued)**

| <b>Field</b> | <b>Description</b>                                                                                                                                               |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 20<br>ERQ11  | Enable DMA Request 11<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 21<br>ERQ10  | Enable DMA Request 10<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled |
| 22<br>ERQ9   | Enable DMA Request 9<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled  |
| 23<br>ERQ8   | Enable DMA Request 8<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled  |
| 24<br>ERQ7   | Enable DMA Request 7<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled  |
| 25<br>ERQ6   | Enable DMA Request 6<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled  |
| 26<br>ERQ5   | Enable DMA Request 5<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled  |
| 27<br>ERQ4   | Enable DMA Request 4<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled  |
| 28<br>ERQ3   | Enable DMA Request 3<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled  |
| 29<br>ERQ2   | Enable DMA Request 2<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled  |
| 30<br>ERQ1   | Enable DMA Request 1<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled  |
| 31<br>ERQ0   | Enable DMA Request 0<br>0 The DMA request signal for the corresponding channel is disabled<br>1 The DMA request signal for the corresponding channel is enabled  |

## 16.4.9 Enable Error Interrupt Register High (DMA\_EEIH)

The EEI{H,L} registers provide a bit map for the 64 channels to enable the error interrupt signal for each channel. EEIH supports channels 63-32, while EEIL covers channels 31-00. The state of any given channel's error interrupt enable is directly affected by writes to this register; it is also affected by writes to the SEEI and CEEI. The {S,C}EEI are provided so the error interrupt enable for a single channel can easily be modified without the need to perform a read-modify-write sequence to the EEI{H,L} registers.

The DMA error indicator and the error interrupt enable flag must be asserted before an error interrupt request for a given channel is asserted to the interrupt controller.

Address: 0h base + 10h offset = 10h

| Bit   | 0     | 1     | 2     | 3     | 4     | 5     | 6     | 7     | 8     | 9     | 10    | 11    | 12    | 13    | 14    | 15    |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| R     | EEI63 | EEI62 | EEI61 | EEI60 | EEI59 | EEI58 | EEI57 | EEI56 | EEI55 | EEI54 | EEI53 | EEI52 | EEI51 | EEI50 | EEI49 | EEI48 |
| W     |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

| Bit   | 16    | 17    | 18    | 19    | 20    | 21    | 22    | 23    | 24    | 25    | 26    | 27    | 28    | 29    | 30    | 31    |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| R     | EEI47 | EEI46 | EEI45 | EEI44 | EEI43 | EEI42 | EEI41 | EEI40 | EEI39 | EEI38 | EEI37 | EEI36 | EEI35 | EEI34 | EEI33 | EEI32 |
| W     |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

### DMA\_EEIH field descriptions

| Field      | Description                                                                                                                                                                                                          |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>EEI63 | Enable Error Interrupt 63<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 1<br>EEI62 | Enable Error Interrupt 62<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 2<br>EEI61 | Enable Error Interrupt 61<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 3<br>EEI60 | Enable Error Interrupt 60<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |

Table continues on the next page...



**DMA\_EEIH field descriptions (continued)**

| <b>Field</b> | <b>Description</b>                                                                                                                                                                                                   |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 4<br>EEI59   | Enable Error Interrupt 59<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 5<br>EEI58   | Enable Error Interrupt 58<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 6<br>EEI57   | Enable Error Interrupt 57<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 7<br>EEI56   | Enable Error Interrupt 56<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 8<br>EEI55   | Enable Error Interrupt 55<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 9<br>EEI54   | Enable Error Interrupt 54<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 10<br>EEI53  | Enable Error Interrupt 53<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 11<br>EEI52  | Enable Error Interrupt 52<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 12<br>EEI51  | Enable Error Interrupt 51<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 13<br>EEI50  | Enable Error Interrupt 50<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 14<br>EEI49  | Enable Error Interrupt 49<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 15<br>EEI48  | Enable Error Interrupt 48<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |

*Table continues on the next page...*

**DMA\_EEIH field descriptions (continued)**

| Field       | Description                                                                                                                                                                                                          |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 16<br>EEI47 | Enable Error Interrupt 47<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 17<br>EEI46 | Enable Error Interrupt 46<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 18<br>EEI45 | Enable Error Interrupt 45<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 19<br>EEI44 | Enable Error Interrupt 44<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 20<br>EEI43 | Enable Error Interrupt 43<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 21<br>EEI42 | Enable Error Interrupt 42<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 22<br>EEI41 | Enable Error Interrupt 41<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 23<br>EEI40 | Enable Error Interrupt 40<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 24<br>EEI39 | Enable Error Interrupt 39<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 25<br>EEI38 | Enable Error Interrupt 38<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 26<br>EEI37 | Enable Error Interrupt 37<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 27<br>EEI36 | Enable Error Interrupt 36<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |

*Table continues on the next page...*

**DMA\_EEIH field descriptions (continued)**

| Field       | Description                                                                                                                                                                                                          |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 28<br>EEI35 | Enable Error Interrupt 35<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 29<br>EEI34 | Enable Error Interrupt 34<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 30<br>EEI33 | Enable Error Interrupt 33<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 31<br>EEI32 | Enable Error Interrupt 32<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |

**16.4.10 Enable Error Interrupt Register Low (DMA\_EEIL)**

The EEIL register provides a bit map for the 64 channels to enable the error interrupt signal for each channel. EEIH supports channels 63-32, while EEIL covers channels 31-00. The state of any given channel's error interrupt enable is directly affected by writes to this register; it is also affected by writes to the SEEI and CEEI. These registers are provided so that the error interrupt enable for a single channel can easily be modified without the need to perform a read-modify-write sequence to the EEIH and EEIL registers.

The DMA error indicator and the error interrupt enable flag must be asserted before an error interrupt request for a given channel is asserted to the interrupt controller.

Address: 0h base + 14h offset = 14h

|       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit   | 0     | 1     | 2     | 3     | 4     | 5     | 6     | 7     | 8     | 9     | 10    | 11    | 12    | 13    | 14    | 15    |
| R     | EEI31 | EEI30 | EEI29 | EEI28 | EEI27 | EEI26 | EEI25 | EEI24 | EEI23 | EEI22 | EEI21 | EEI20 | EEI19 | EEI18 | EEI17 | EEI16 |
| W     |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| Bit   | 16    | 17    | 18    | 19    | 20    | 21    | 22    | 23    | 24    | 25    | 26    | 27    | 28    | 29    | 30    | 31    |
| R     | EEI15 | EEI14 | EEI13 | EEI12 | EEI11 | EEI10 | EEI9  | EEI8  | EEI7  | EEI6  | EEI5  | EEI4  | EEI3  | EEI2  | EEI1  | EEI0  |
| W     |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

**DMA\_EEIL field descriptions**

| Field       | Description                                                                                                                                                                                                          |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>EEI31  | Enable Error Interrupt 31<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 1<br>EEI30  | Enable Error Interrupt 30<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 2<br>EEI29  | Enable Error Interrupt 29<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 3<br>EEI28  | Enable Error Interrupt 28<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 4<br>EEI27  | Enable Error Interrupt 27<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 5<br>EEI26  | Enable Error Interrupt 26<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 6<br>EEI25  | Enable Error Interrupt 25<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 7<br>EEI24  | Enable Error Interrupt 24<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 8<br>EEI23  | Enable Error Interrupt 23<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 9<br>EEI22  | Enable Error Interrupt 22<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 10<br>EEI21 | Enable Error Interrupt 21<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 11<br>EEI20 | Enable Error Interrupt 20<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |

*Table continues on the next page...*

**DMA\_EEIL field descriptions (continued)**

| <b>Field</b> | <b>Description</b>                                                                                                                                                                                                   |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 12<br>EEI19  | Enable Error Interrupt 19<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 13<br>EEI18  | Enable Error Interrupt 18<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 14<br>EEI17  | Enable Error Interrupt 17<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 15<br>EEI16  | Enable Error Interrupt 16<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 16<br>EEI15  | Enable Error Interrupt 15<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 17<br>EEI14  | Enable Error Interrupt 14<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 18<br>EEI13  | Enable Error Interrupt 13<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 19<br>EEI12  | Enable Error Interrupt 12<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 20<br>EEI11  | Enable Error Interrupt 11<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 21<br>EEI10  | Enable Error Interrupt 10<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 22<br>EEI9   | Enable Error Interrupt 9<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request  |
| 23<br>EEI8   | Enable Error Interrupt 8<br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request  |

*Table continues on the next page...*

**DMA\_EEIL field descriptions (continued)**

| <b>Field</b> | <b>Description</b>                                                                                                                                                                                                      |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 24<br>EEI7   | Enable Error Interrupt 7<br><br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 25<br>EEI6   | Enable Error Interrupt 6<br><br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 26<br>EEI5   | Enable Error Interrupt 5<br><br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 27<br>EEI4   | Enable Error Interrupt 4<br><br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 28<br>EEI3   | Enable Error Interrupt 3<br><br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 29<br>EEI2   | Enable Error Interrupt 2<br><br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 30<br>EEI1   | Enable Error Interrupt 1<br><br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |
| 31<br>EEI0   | Enable Error Interrupt 0<br><br>0 The error signal for corresponding channel does not generate an error interrupt<br>1 The assertion of the error signal for corresponding channel generates an error interrupt request |

### 16.4.11 Set Enable Request Register (DMA\_SERQ)

The SERQ provides a simple memory-mapped mechanism to set a given bit in the ERQ {H,L} to enable the DMA request for a given channel. The data value on a register write causes the corresponding bit in the ERQ {H,L} to be set. Setting the SAER bit provides a global set function, forcing the entire contents of ERQ {H,L} to be set. If the NOP bit is set, the command is ignored. This allows you to write multiple-byte registers as a 32-bit word. Reads of this register return all zeroes.

Address: 0h base + 18h offset = 18h

|       |     |      |      |   |   |   |   |   |
|-------|-----|------|------|---|---|---|---|---|
| Bit   | 0   | 1    | 2    | 3 | 4 | 5 | 6 | 7 |
| Read  | 0   | 0    | 0    |   |   |   |   |   |
| Write | NOP | SAER | SERQ |   |   |   |   |   |
| Reset | 0   | 0    | 0    | 0 | 0 | 0 | 0 | 0 |

**DMA\_SERQ field descriptions**

| Field       | Description                                                                                                            |
|-------------|------------------------------------------------------------------------------------------------------------------------|
| 0<br>NOP    | No Op enable<br><br>0 Normal operation<br>1 No operation, ignore the other bits in this register                       |
| 1<br>SAER   | Set All Enable Requests<br><br>0 Set only the ERQ {H,L} bit specified in the SERQ field<br>1 Set all bits in ERQ {H,L} |
| 2–7<br>SERQ | Set Enable Request<br><br>Sets the corresponding bit in ERQ{H,L}.                                                      |

### 16.4.12 Clear Enable Request Register (DMA\_CERQ)

The CERQ provides a simple memory-mapped mechanism to clear a given bit in the ERQ {H,L} to disable the DMA request for a given channel. The data value on a register write causes the corresponding bit in the ERQ {H,L} to be cleared. Setting the CAER bit provides a global clear function, forcing the entire contents of the ERQ {H,L} to be cleared, disabling all DMA request inputs. If NOP is set, the command is ignored. This allows you to write multiple-byte registers as a 32-bit word. Reads of this register return all zeroes.

NOTE

Disable a channel’s hardware service request at the source before clearing the channel’s ERQ bit.

Address: 0h base + 19h offset = 19h

|       |     |      |      |   |   |   |   |   |
|-------|-----|------|------|---|---|---|---|---|
| Bit   | 0   | 1    | 2    | 3 | 4 | 5 | 6 | 7 |
| Read  | 0   | 0    | 0    |   |   |   |   |   |
| Write | NOP | CAER | CERQ |   |   |   |   |   |
| Reset | 0   | 0    | 0    | 0 | 0 | 0 | 0 | 0 |

DMA\_CERQ field descriptions

| Field       | Description                                                                                                                  |
|-------------|------------------------------------------------------------------------------------------------------------------------------|
| 0<br>NOP    | No Op enable<br><br>0 Normal operation<br>1 No operation, ignore the other bits in this register                             |
| 1<br>CAER   | Clear All Enable Requests<br><br>0 Clear only the ERQ {H,L} bit specified in the CERQ field<br>1 Clear all bits in ERQ {H,L} |
| 2–7<br>CERQ | Clear Enable Request<br><br>Clears the corresponding bit in ERQ{H,L}.                                                        |

16.4.13 Set Enable Error Interrupt Register (DMA\_SEEI)

The SEEI provides a simple memory-mapped mechanism to set a given bit in the EEI {H,L} to enable the error interrupt for a given channel. The data value on a register write causes the corresponding bit in the EEI {H,L} to be set. Setting the SAEE bit provides a global set function, forcing the entire EEI {H,L} contents to be set. If the NOP bit is set, the command is ignored. This allows you to write multiple-byte registers as a 32-bit word. Reads of this register return all zeroes.

Address: 0h base + 1Ah offset = 1Ah

|       |     |      |      |   |   |   |   |   |
|-------|-----|------|------|---|---|---|---|---|
| Bit   | 0   | 1    | 2    | 3 | 4 | 5 | 6 | 7 |
| Read  | 0   | 0    | 0    |   |   |   |   |   |
| Write | NOP | SAEE | SEEI |   |   |   |   |   |
| Reset | 0   | 0    | 0    | 0 | 0 | 0 | 0 | 0 |



**DMA\_SEEI field descriptions**

| Field       | Description                                                                                                                       |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------|
| 0<br>NOP    | No Op enable<br><br>0 Normal operation<br>1 No operation, ignore the other bits in this register                                  |
| 1<br>SAEE   | Sets All Enable Error Interrupts<br><br>0 Set only the EEI {H,L} bit specified in the SEEI field.<br>1 Sets all bits in EEI {H,L} |
| 2–7<br>SEEI | Set Enable Error Interrupt<br><br>Sets the corresponding bit in EEI {H,L}                                                         |

**16.4.14 Clear Enable Error Interrupt Register (DMA\_CEEI)**

The CEEI provides a simple memory-mapped mechanism to clear a given bit in the EEI {H,L} to disable the error interrupt for a given channel. The data value on a register write causes the corresponding bit in the EEI {H,L} to be cleared. Setting the CAEE bit provides a global clear function, forcing the EEI {H,L} contents to be cleared, disabling all DMA request inputs. If the NOP bit is set, the command is ignored. This allows you to write multiple-byte registers as a 32-bit word. Reads of this register return all zeroes.

Address: 0h base + 1Bh offset = 1Bh

|       |     |      |      |   |   |   |   |   |
|-------|-----|------|------|---|---|---|---|---|
| Bit   | 0   | 1    | 2    | 3 | 4 | 5 | 6 | 7 |
| Read  | 0   | 0    | 0    |   |   |   |   |   |
| Write | NOP | CAEE | CEEI |   |   |   |   |   |
| Reset | 0   | 0    | 0    | 0 | 0 | 0 | 0 | 0 |

**DMA\_CEEI field descriptions**

| Field       | Description                                                                                                                          |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>NOP    | No Op enable<br><br>0 Normal operation<br>1 No operation, ignore the other bits in this register                                     |
| 1<br>CAEE   | Clear All Enable Error Interrupts<br><br>0 Clear only the EEI {H,L} bit specified in the CEEI field<br>1 Clear all bits in EEI {H,L} |
| 2–7<br>CEEI | Clear Enable Error Interrupt<br><br>Clears the corresponding bit in EEI{H,L}                                                         |

16.4.15 Clear Interrupt Request Register (DMA\_CINT)

The CINT provides a simple, memory-mapped mechanism to clear a given bit in the INT {H,L} to disable the interrupt request for a given channel. The given value on a register write causes the corresponding bit in the INT {H,L} to be cleared. Setting the CAIR bit provides a global clear function, forcing the entire contents of the INT {H,L} to be cleared, disabling all DMA interrupt requests. If the NOP bit is set, the command is ignored. This allows you to write multiple-byte registers as a 32-bit word. Reads of this register return all zeroes.

Address: 0h base + 1Ch offset = 1Ch

|       |     |      |      |   |   |   |   |   |
|-------|-----|------|------|---|---|---|---|---|
| Bit   | 0   | 1    | 2    | 3 | 4 | 5 | 6 | 7 |
| Read  | 0   | 0    | 0    |   |   |   |   |   |
| Write | NOP | CAIR | CINT |   |   |   |   |   |
| Reset | 0   | 0    | 0    | 0 | 0 | 0 | 0 | 0 |

DMA\_CINT field descriptions

| Field       | Description                                                                                                                 |
|-------------|-----------------------------------------------------------------------------------------------------------------------------|
| 0<br>NOP    | No Op enable<br>0 Normal operation<br>1 No operation, ignore the other bits in this register                                |
| 1<br>CAIR   | Clear All Interrupt Requests<br>0 Clear only the INT {H,L} bit specified in the CINT field<br>1 Clear all bits in INT {H,L} |
| 2-7<br>CINT | Clear Interrupt Request<br>Clears the corresponding bit in INT {H,L}                                                        |

### 16.4.16 Clear Error Register (DMA\_CERR)

The CERR provides a simple memory-mapped mechanism to clear a given bit in the ERR {H,L} to disable the error condition flag for a given channel. The given value on a register write causes the corresponding bit in the ERR {H,L} to be cleared. Setting the CAEI bit provides a global clear function, forcing the ERR {H,L} contents to be cleared, clearing all channel error indicators. If the NOP bit is set, the command is ignored. This allows you to write multiple-byte registers as a 32-bit word. Reads of this register return all zeroes.

Address: 0h base + 1Dh offset = 1Dh

|       |     |      |      |   |   |   |   |   |
|-------|-----|------|------|---|---|---|---|---|
| Bit   | 0   | 1    | 2    | 3 | 4 | 5 | 6 | 7 |
| Read  | 0   | 0    | 0    |   |   |   |   |   |
| Write | NOP | CAEI | CERR |   |   |   |   |   |
| Reset | 0   | 0    | 0    | 0 | 0 | 0 | 0 | 0 |

DMA\_CERR field descriptions

| Field       | Description                                                                                                               |
|-------------|---------------------------------------------------------------------------------------------------------------------------|
| 0<br>NOP    | No Op enable<br>0 Normal operation<br>1 No operation, ignore the other bits in this register                              |
| 1<br>CAEI   | Clear All Error Indicators<br>0 Clear only the ERR {H,L} bit specified in the CERR field<br>1 Clear all bits in ERR {H,L} |
| 2–7<br>CERR | Clear Error Indicator<br>Clears the corresponding bit in ERR {H,L}                                                        |

16.4.17 Set START Bit Register (DMA\_SSRT)

The SSRT provides a simple memory-mapped mechanism to set the START bit in the TCD of the given channel. The data value on a register write causes the START bit in the corresponding transfer control descriptor to be set. Setting the SAST bit provides a global set function, forcing all START bits to be set. If the NOP bit is set, the command is ignored. This allows you to write multiple-byte registers as a 32-bit word. Reads of this register return all zeroes.

Address: 0h base + 1Eh offset = 1Eh

|       |     |      |      |   |   |   |   |   |
|-------|-----|------|------|---|---|---|---|---|
| Bit   | 0   | 1    | 2    | 3 | 4 | 5 | 6 | 7 |
| Read  | 0   | 0    | 0    |   |   |   |   |   |
| Write | NOP | SAST | SSRT |   |   |   |   |   |
| Reset | 0   | 0    | 0    | 0 | 0 | 0 | 0 | 0 |

DMA\_SSRT field descriptions

| Field       | Description                                                                                                                                            |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>NOP    | No Op enable<br><br>0 Normal operation<br>1 No operation, ignore the other bits in this register                                                       |
| 1<br>SAST   | Set All START Bits (activates all channels)<br><br>0 Set only the TCDn_CSR[START] bit specified in the SSRT field<br>1 Set all bits in TCDn_CSR[START] |
| 2–7<br>SSRT | Set START Bit<br><br>Sets the corresponding bit in TCDn_CSR[START]                                                                                     |

### 16.4.18 Clear DONE Status Bit Register (DMA\_CDNE)

The CDNE provides a simple memory-mapped mechanism to clear the DONE bit in the TCD of the given channel. The data value on a register write causes the DONE bit in the corresponding transfer control descriptor to be cleared. Setting the CADN bit provides a global clear function, forcing all DONE bits to be cleared. If the NOP bit is set, the command is ignored. This allows you to write multiple-byte registers as a 32-bit word. Reads of this register return all zeroes.

Address: 0h base + 1Fh offset = 1Fh

|       |     |      |      |   |   |   |   |   |
|-------|-----|------|------|---|---|---|---|---|
| Bit   | 0   | 1    | 2    | 3 | 4 | 5 | 6 | 7 |
| Read  | 0   | 0    | 0    |   |   |   |   |   |
| Write | NOP | CADN | CDNE |   |   |   |   |   |
| Reset | 0   | 0    | 0    | 0 | 0 | 0 | 0 | 0 |

DMA\_CDNE field descriptions

| Field       | Description                                                                                                                         |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>NOP    | No Op enable<br><br>0 Normal operation<br>1 No operation, ignore the other bits in this register                                    |
| 1<br>CADN   | Clears All DONE Bits<br><br>0 Clears only the TCDn_CSR[DONE] bit specified in the CDNE field<br>1 Clears all bits in TCDn_CSR[DONE] |
| 2–7<br>CDNE | Clear DONE Bit<br><br>Clears the corresponding bit in TCDn_CSR[DONE]                                                                |

### 16.4.19 Interrupt Request Register High (DMA\_INTH)

The INT{H,L} provide a bit map for the 64 channels signaling the presence of an interrupt request for each channel. Depending on the appropriate bit setting in the transfer-control descriptions, the eDMA engine generates an interrupt on a data transfer completion. The outputs of this register are directly routed to the interrupt controller. During the interrupt-service routine associated with any given channel, the software must clear the appropriate bit, negating the interrupt request. Typically, a write to the CINT register in the interrupt service routine is used for this purpose.

## Memory map/register definition

The state of any given channel's interrupt request is directly affected by writes to this register; it is also affected by writes to the CINT register. On writes to INT, a 1 in any bit position clears the corresponding channel's interrupt request. A zero in any bit position has no affect on the corresponding channel's current interrupt status. The CINT register is provided so the interrupt request for a single channel can easily be cleared without the need to perform a read-modify-write sequence to the INT{H,L} register.

Address: 0h base + 20h offset = 20h

| Bit   | 0     | 1     | 2     | 3     | 4     | 5     | 6     | 7     | 8     | 9     | 10    | 11    | 12    | 13    | 14    | 15    |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| R     | INT63 | INT62 | INT61 | INT60 | INT59 | INT58 | INT57 | INT56 | INT55 | INT54 | INT53 | INT52 | INT51 | INT50 | INT49 | INT48 |
| W     | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

| Bit   | 16    | 17    | 18    | 19    | 20    | 21    | 22    | 23    | 24    | 25    | 26    | 27    | 28    | 29    | 30    | 31    |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| R     | INT47 | INT46 | INT45 | INT44 | INT43 | INT42 | INT41 | INT40 | INT39 | INT38 | INT37 | INT36 | INT35 | INT34 | INT33 | INT32 |
| W     | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

## DMA\_INTH field descriptions

| Field      | Description                                                                                                                                         |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>INT63 | Interrupt Request 63<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 1<br>INT62 | Interrupt Request 62<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 2<br>INT61 | Interrupt Request 61<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 3<br>INT60 | Interrupt Request 60<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 4<br>INT59 | Interrupt Request 59                                                                                                                                |

Table continues on the next page...

**DMA\_INTH field descriptions (continued)**

| <b>Field</b> | <b>Description</b>                                                                                                                                  |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
|              | 0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active                         |
| 5<br>INT58   | Interrupt Request 58<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 6<br>INT57   | Interrupt Request 57<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 7<br>INT56   | Interrupt Request 56<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 8<br>INT55   | Interrupt Request 55<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 9<br>INT54   | Interrupt Request 54<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 10<br>INT53  | Interrupt Request 53<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 11<br>INT52  | Interrupt Request 52<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 12<br>INT51  | Interrupt Request 51<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 13<br>INT50  | Interrupt Request 50<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 14<br>INT49  | Interrupt Request 49<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 15<br>INT48  | Interrupt Request 48<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 16<br>INT47  | Interrupt Request 47                                                                                                                                |

*Table continues on the next page...*

**DMA\_INTH field descriptions (continued)**

| Field       | Description                                                                                                                                         |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
|             | 0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active                         |
| 17<br>INT46 | Interrupt Request 46<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 18<br>INT45 | Interrupt Request 45<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 19<br>INT44 | Interrupt Request 44<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 20<br>INT43 | Interrupt Request 43<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 21<br>INT42 | Interrupt Request 42<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 22<br>INT41 | Interrupt Request 41<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 23<br>INT40 | Interrupt Request 40<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 24<br>INT39 | Interrupt Request 30<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 25<br>INT38 | Interrupt Request 38<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 26<br>INT37 | Interrupt Request 37<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 27<br>INT36 | Interrupt Request 36<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 28<br>INT35 | Interrupt Request 35                                                                                                                                |

*Table continues on the next page...*



**DMA\_INTL field descriptions (continued)**

| Field       | Description                                                                                                                                         |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
|             | 0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active                         |
| 29<br>INT34 | Interrupt Request 34<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 30<br>INT33 | Interrupt Request 33<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 31<br>INT32 | Interrupt Request 32<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |

**16.4.20 Interrupt Request Register Low (DMA\_INTL)**

The INT {H,L} register provides a bit map for the 64 channels signaling the presence of an interrupt request for each channel. Depending on the appropriate bit setting in the transfer-control descriptors, the eDMA engine generates an interrupt on data transfer completion. The outputs of this register are directly routed to the interrupt controller. During the interrupt-service routine associated with any given channel, it is the software's responsibility to clear the appropriate bit, negating the interrupt request. Typically, a write to the CINT register in the interrupt service routine is used for this purpose.

The state of any given channel's interrupt request is directly affected by writes to this register; it is also affected by writes to the CINT register. On writes to INT, a 1 in any bit position clears the corresponding channel's interrupt request. A zero in any bit position has no affect on the corresponding channel's current interrupt status. The CINT register is provided so the interrupt request for a single channel can easily be cleared without the need to perform a read-modify-write sequence to the INT {H,L} register.

Address: 0h base + 24h offset = 24h

| Bit   | 0     | 1     | 2     | 3     | 4     | 5     | 6     | 7     | 8     | 9     | 10    | 11    | 12    | 13    | 14    | 15    |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| R     | INT31 | INT30 | INT29 | INT28 | INT27 | INT26 | INT25 | INT24 | INT23 | INT22 | INT21 | INT20 | INT19 | INT18 | INT17 | INT16 |
| W     | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

## Memory map/register definition

| Bit   | 16    | 17    | 18    | 19    | 20    | 21    | 22   | 23   | 24   | 25   | 26   | 27   | 28   | 29   | 30   | 31   |
|-------|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| R     | INT15 | INT14 | INT13 | INT12 | INT11 | INT10 | INT9 | INT8 | INT7 | INT6 | INT5 | INT4 | INT3 | INT2 | INT1 | INT0 |
| W     | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c  | w1c  | w1c  | w1c  | w1c  | w1c  | w1c  | w1c  | w1c  | w1c  |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |

### DMA\_INTL field descriptions

| Field      | Description                                                                                                                                         |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>INT31 | Interrupt Request 31<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 1<br>INT30 | Interrupt Request 30<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 2<br>INT29 | Interrupt Request 29<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 3<br>INT28 | Interrupt Request 28<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 4<br>INT27 | Interrupt Request 27<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 5<br>INT26 | Interrupt Request 26<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 6<br>INT25 | Interrupt Request 25<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 7<br>INT24 | Interrupt Request 24<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 8<br>INT23 | Interrupt Request 23<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 9<br>INT22 | Interrupt Request 22                                                                                                                                |

Table continues on the next page...

**DMA\_INTL field descriptions (continued)**

| <b>Field</b> | <b>Description</b>                                                                                                                                  |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
|              | 0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active                         |
| 10<br>INT21  | Interrupt Request 21<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 11<br>INT20  | Interrupt Request 20<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 12<br>INT19  | Interrupt Request 19<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 13<br>INT18  | Interrupt Request 18<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 14<br>INT17  | Interrupt Request 17<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 15<br>INT16  | Interrupt Request 16<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 16<br>INT15  | Interrupt Request 15<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 17<br>INT14  | Interrupt Request 14<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 18<br>INT13  | Interrupt Request 13<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 19<br>INT12  | Interrupt Request 12<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 20<br>INT11  | Interrupt Request 11<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 21<br>INT10  | Interrupt Request 10                                                                                                                                |

*Table continues on the next page...*

**DMA\_INTL field descriptions (continued)**

| <b>Field</b> | <b>Description</b>                                                                                                                                 |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
|              | 0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active                        |
| 22<br>INT9   | Interrupt Request 9<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 23<br>INT8   | Interrupt Request 8<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 24<br>INT7   | Interrupt Request 7<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 25<br>INT6   | Interrupt Request 6<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 26<br>INT5   | Interrupt Request 5<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 27<br>INT4   | Interrupt Request 4<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 28<br>INT3   | Interrupt Request 3<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 29<br>INT2   | Interrupt Request 2<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 30<br>INT1   | Interrupt Request 1<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |
| 31<br>INT0   | Interrupt Request 0<br>0 The interrupt request for corresponding channel is cleared<br>1 The interrupt request for corresponding channel is active |

### 16.4.21 Error Register High (DMA\_ERRH)

The ERR{H,L} provide a bit map for the 64 channels, signaling the presence of an error for each channel. ERRH supports channels 63-32, while ERRL covers channels 31-00. The eDMA engine signals the occurrence of a error condition by setting the appropriate bit in this register. The outputs of this register are enabled by the contents of EEI, then logically summed across groups of 16, 32 and 64 channels to form several group error interrupt requests. These are then routed to the interrupt controller. During the execution of the interrupt-service routine associated with any DMA errors, the software must clear the appropriate bit, negating the error-interrupt request. Typically, a write to CERR in the interrupt-service routine is used for this purpose. The normal DMA channel completion indicators, setting the transfer control descriptor DONE flag and the possible assertion of an interrupt request, are not affected when an error is detected.

The contents of this register can also be polled, because a non-zero value indicates the presence of a channel error regardless of the state of EEI. The state of any given channel's error indicators is affected by writes to this register; it is also affected by writes to CERR. On writes to ERR, a one in any bit position clears the corresponding channel's error status. A zero in any bit position has no affect on the corresponding channel's current error status. CERR is provided so the error indicator for a single channel can easily be cleared.

Address: 0h base + 28h offset = 28h

| Bit   | 0     | 1     | 2     | 3     | 4     | 5     | 6     | 7     | 8     | 9     | 10    | 11    | 12    | 13    | 14    | 15    |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| R     | ERR63 | ERR62 | ERR61 | ERR60 | ERR59 | ERR58 | ERR57 | ERR56 | ERR55 | ERR54 | ERR53 | ERR52 | ERR51 | ERR50 | ERR49 | ERR48 |
| W     | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

| Bit   | 16    | 17    | 18    | 19    | 20    | 21    | 22    | 23    | 24    | 25    | 26    | 27    | 28    | 29    | 30    | 31    |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| R     | ERR47 | ERR46 | ERR45 | ERR44 | ERR43 | ERR42 | ERR41 | ERR40 | ERR39 | ERR38 | ERR37 | ERR36 | ERR35 | ERR34 | ERR33 | ERR32 |
| W     | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

**DMA\_ERRH field descriptions**

| Field       | Description                                                                                                   |
|-------------|---------------------------------------------------------------------------------------------------------------|
| 0<br>ERR63  | Error In Channel 63<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 1<br>ERR62  | Error In Channel 62<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 2<br>ERR61  | Error In Channel 61<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 3<br>ERR60  | Error In Channel 60<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 4<br>ERR59  | Error In Channel 59<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 5<br>ERR58  | Error In Channel 58<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 6<br>ERR57  | Error In Channel 57<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 7<br>ERR56  | Error In Channel 56<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 8<br>ERR55  | Error In Channel 55<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 9<br>ERR54  | Error In Channel 54<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 10<br>ERR53 | Error In Channel 53<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 11<br>ERR52 | Error In Channel 52<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |

*Table continues on the next page...*

**DMA\_ERRH field descriptions (continued)**

| <b>Field</b> | <b>Description</b>                                                                                            |
|--------------|---------------------------------------------------------------------------------------------------------------|
| 12<br>ERR51  | Error In Channel 51<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 13<br>ERR50  | Error In Channel 50<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 14<br>ERR49  | Error In Channel 49<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 15<br>ERR48  | Error In Channel 48<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 16<br>ERR47  | Error In Channel 47<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 17<br>ERR46  | Error In Channel 46<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 18<br>ERR45  | Error In Channel 45<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 19<br>ERR44  | Error In Channel 44<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 20<br>ERR43  | Error In Channel 43<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 21<br>ERR42  | Error In Channel 42<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 22<br>ERR41  | Error In Channel 41<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 23<br>ERR40  | Error In Channel 40<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |

*Table continues on the next page...*

**DMA\_ERRH field descriptions (continued)**

| Field       | Description                                                                                                   |
|-------------|---------------------------------------------------------------------------------------------------------------|
| 24<br>ERR39 | Error In Channel 39<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 25<br>ERR38 | Error In Channel 38<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 26<br>ERR37 | Error In Channel 37<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 27<br>ERR36 | Error In Channel 36<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 28<br>ERR35 | Error In Channel 35<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 29<br>ERR34 | Error In Channel 34<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 30<br>ERR33 | Error In Channel 33<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 31<br>ERR32 | Error In Channel 32<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |



### 16.4.22 Error Register Low (DMA\_ERRL)

The ERR {H,L} provides a bit map for the 64 channels, signaling the presence of an error for each channel. ERRH supports channels 63-32, while ERRL covers channels 31-00. The eDMA engine signals the occurrence of an error condition by setting the appropriate bit in this register. The outputs of this register are enabled by the contents of the EEI, then logically summed across groups of 16, 32 and 64 channels to form several group error interrupt requests, which are then routed to the interrupt controller. During the execution of the interrupt-service routine associated with any DMA errors, it is software's responsibility to clear the appropriate bit, negating the error-interrupt request. Typically, a write to the CERR in the interrupt-service routine is used for this purpose. The normal DMA channel completion indicators (setting the transfer control descriptor DONE flag and the possible assertion of an interrupt request) are not affected when an error is detected.

The contents of this register can also be polled because a non-zero value indicates the presence of a channel error regardless of the state of the EEI. The state of any given channel's error indicators is affected by writes to this register; it is also affected by writes to the CERR. On writes to the ERR, a one in any bit position clears the corresponding channel's error status. A zero in any bit position has no affect on the corresponding channel's current error status. The CERR is provided so the error indicator for a single channel can easily be cleared.

Address: 0h base + 2Ch offset = 2Ch

| Bit   | 0     | 1     | 2     | 3     | 4     | 5     | 6     | 7     | 8     | 9     | 10    | 11    | 12    | 13    | 14    | 15    |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| R     | ERR31 | ERR30 | ERR29 | ERR28 | ERR27 | ERR26 | ERR25 | ERR24 | ERR23 | ERR22 | ERR21 | ERR20 | ERR19 | ERR18 | ERR17 | ERR16 |
| W     | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

| Bit   | 16    | 17    | 18    | 19    | 20    | 21    | 22   | 23   | 24   | 25   | 26   | 27   | 28   | 29   | 30   | 31   |
|-------|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| R     | ERR15 | ERR14 | ERR13 | ERR12 | ERR11 | ERR10 | ERR9 | ERR8 | ERR7 | ERR6 | ERR5 | ERR4 | ERR3 | ERR2 | ERR1 | ERR0 |
| W     | w1c   | w1c   | w1c   | w1c   | w1c   | w1c   | w1c  | w1c  | w1c  | w1c  | w1c  | w1c  | w1c  | w1c  | w1c  | w1c  |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |

**DMA\_ERRL field descriptions**

| Field       | Description                                                                                                   |
|-------------|---------------------------------------------------------------------------------------------------------------|
| 0<br>ERR31  | Error In Channel 31<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 1<br>ERR30  | Error In Channel 30<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 2<br>ERR29  | Error In Channel 29<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 3<br>ERR28  | Error In Channel 28<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 4<br>ERR27  | Error In Channel 27<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 5<br>ERR26  | Error In Channel 26<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 6<br>ERR25  | Error In Channel 25<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 7<br>ERR24  | Error In Channel 24<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 8<br>ERR23  | Error In Channel 23<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 9<br>ERR22  | Error In Channel 22<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 10<br>ERR21 | Error In Channel 21<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 11<br>ERR20 | Error In Channel 20<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |

*Table continues on the next page...*

**DMA\_ERRL field descriptions (continued)**

| <b>Field</b> | <b>Description</b>                                                                                            |
|--------------|---------------------------------------------------------------------------------------------------------------|
| 12<br>ERR19  | Error In Channel 19<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 13<br>ERR18  | Error In Channel 18<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 14<br>ERR17  | Error In Channel 17<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 15<br>ERR16  | Error In Channel 16<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 16<br>ERR15  | Error In Channel 15<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 17<br>ERR14  | Error In Channel 14<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 18<br>ERR13  | Error In Channel 13<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 19<br>ERR12  | Error In Channel 12<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 20<br>ERR11  | Error In Channel 11<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 21<br>ERR10  | Error In Channel 10<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 22<br>ERR9   | Error In Channel 9<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred  |
| 23<br>ERR8   | Error In Channel 8<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred  |

*Table continues on the next page...*

**DMA\_ERRL field descriptions (continued)**

| Field      | Description                                                                                                  |
|------------|--------------------------------------------------------------------------------------------------------------|
| 24<br>ERR7 | Error In Channel 7<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 25<br>ERR6 | Error In Channel 6<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 26<br>ERR5 | Error In Channel 5<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 27<br>ERR4 | Error In Channel 4<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 28<br>ERR3 | Error In Channel 3<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 29<br>ERR2 | Error In Channel 2<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 30<br>ERR1 | Error In Channel 1<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |
| 31<br>ERR0 | Error In Channel 0<br>0 An error in this channel has not occurred<br>1 An error in this channel has occurred |

**16.4.23 Hardware Request Status Register High (DMA\_HRSH)**

The HRS{H,L} provide a bit map for the DMA channels, signaling the presence of a hardware request for each channel. HRSH supports channels 63-32, while HRSL covers channels 31-0. The hardware request status bits reflect the current state of the register and DMA request signals, qualified via the ERQ{H,L} fields, as seen by the DMA's arbitration logic. This view into the hardware request signals may be used for debug purposes.

**NOTE**

These bits reflect the state of the request as seen by the arbitration logic. Therefore, this status is affected by the ERQ {H,L} bits.

The HRS bit for its respective channel remains asserted for the period when a Hardware Request is Present on the Channel. After the Request is completed and Channel is free, the HRS bit is automatically cleared by hardware.

Address: 0h base + 30h offset = 30h

|       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit   | 0     | 1     | 2     | 3     | 4     | 5     | 6     | 7     | 8     | 9     | 10    | 11    | 12    | 13    | 14    | 15    |
| R     | HRS63 | HRS62 | HRS61 | HRS60 | HRS59 | HRS58 | HRS57 | HRS56 | HRS55 | HRS54 | HRS53 | HRS52 | HRS51 | HRS50 | HRS49 | HRS48 |
| W     |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

|       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit   | 16    | 17    | 18    | 19    | 20    | 21    | 22    | 23    | 24    | 25    | 26    | 27    | 28    | 29    | 30    | 31    |
| R     | HRS47 | HRS46 | HRS45 | HRS44 | HRS43 | HRS42 | HRS41 | HRS40 | HRS39 | HRS38 | HRS37 | HRS36 | HRS35 | HRS34 | HRS33 | HRS32 |
| W     |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

**DMA\_HRSH field descriptions**

| Field      | Description                                                                                                                                                                              |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>HRS63 | Hardware Request Status Channel 63<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 1<br>HRS62 | Hardware Request Status Channel 62<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |

*Table continues on the next page...*

**DMA\_HRSH field descriptions (continued)**

| Field       | Description                                                                                                                                                                              |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2<br>HRS61  | Hardware Request Status Channel 61<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 3<br>HRS60  | Hardware Request Status Channel 60<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 4<br>HRS59  | Hardware Request Status Channel 59<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 5<br>HRS58  | Hardware Request Status Channel 58<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 6<br>HRS57  | Hardware Request Status Channel 57<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 7<br>HRS56  | Hardware Request Status Channel 56<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 8<br>HRS55  | Hardware Request Status Channel 55<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 9<br>HRS54  | Hardware Request Status Channel 54<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 10<br>HRS53 | Hardware Request Status Channel 53<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 11<br>HRS52 | Hardware Request Status Channel 52<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 12<br>HRS51 | Hardware Request Status Channel 51<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 13<br>HRS50 | Hardware Request Status Channel 50<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |

*Table continues on the next page...*

**DMA\_HRS field descriptions (continued)**

| <b>Field</b> | <b>Description</b>                                                                                                                                                                       |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 14<br>HRS49  | Hardware Request Status Channel 49<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 15<br>HRS48  | Hardware Request Status Channel 48<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 16<br>HRS47  | Hardware Request Status Channel 47<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 17<br>HRS46  | Hardware Request Status Channel 46<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 18<br>HRS45  | Hardware Request Status Channel 45<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 19<br>HRS44  | Hardware Request Status Channel 44<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 20<br>HRS43  | Hardware Request Status Channel 43<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 21<br>HRS42  | Hardware Request Status Channel 42<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 22<br>HRS41  | Hardware Request Status Channel 41<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 23<br>HRS40  | Hardware Request Status Channel 40<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 24<br>HRS39  | Hardware Request Status Channel 39<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 25<br>HRS38  | Hardware Request Status Channel 38<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |

*Table continues on the next page...*

**DMA\_HRSH field descriptions (continued)**

| Field       | Description                                                                                                                                                                              |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 26<br>HRS37 | Hardware Request Status Channel 37<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 27<br>HRS36 | Hardware Request Status Channel 36<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 28<br>HRS35 | Hardware Request Status Channel 35<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 29<br>HRS34 | Hardware Request Status Channel 34<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 30<br>HRS33 | Hardware Request Status Channel 33<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 31<br>HRS32 | Hardware Request Status Channel 32<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |

**16.4.24 Hardware Request Status Register Low (DMA\_HRSL)**

The HRS{H,L} registers provide a bit map for the DMA channels, signaling the presence of a hardware request for each channel. HRSH supports channels 63-32, while HRSL covers channels 31-0. The hardware request status bits reflect the current state of the register and qualified (via the ERQ {H,L} fields) DMA request signals as seen by the DMA's arbitration logic. This view into the hardware request signals may be used for debug purposes.

**NOTE**

These bits reflect the state of the request as seen by the arbitration logic. Therefore, this status is affected by the ERQ {H,L} bits.

The HRS bit for its respective channel remains asserted for the period when a Hardware Request is Present on the Channel. After the Request is completed and Channel is free, the HRS bit is automatically cleared by hardware.



Address: 0h base + 34h offset = 34h

|       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit   | 0     | 1     | 2     | 3     | 4     | 5     | 6     | 7     | 8     | 9     | 10    | 11    | 12    | 13    | 14    | 15    |
| R     | HRS31 | HRS30 | HRS29 | HRS28 | HRS27 | HRS26 | HRS25 | HRS24 | HRS23 | HRS22 | HRS21 | HRS20 | HRS19 | HRS18 | HRS17 | HRS16 |
| W     |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

|       |       |       |       |       |       |       |      |      |      |      |      |      |      |      |      |      |
|-------|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| Bit   | 16    | 17    | 18    | 19    | 20    | 21    | 22   | 23   | 24   | 25   | 26   | 27   | 28   | 29   | 30   | 31   |
| R     | HRS15 | HRS14 | HRS13 | HRS12 | HRS11 | HRS10 | HRS9 | HRS8 | HRS7 | HRS6 | HRS5 | HRS4 | HRS3 | HRS2 | HRS1 | HRS0 |
| W     |       |       |       |       |       |       |      |      |      |      |      |      |      |      |      |      |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |

**DMA\_HRSL field descriptions**

| Field      | Description                                                                                                                                                                              |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>HRS31 | Hardware Request Status Channel 31<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 1<br>HRS30 | Hardware Request Status Channel 30<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 2<br>HRS29 | Hardware Request Status Channel 29<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 3<br>HRS28 | Hardware Request Status Channel 28<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 4<br>HRS27 | Hardware Request Status Channel 27<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |

*Table continues on the next page...*

**DMA\_HRSL field descriptions (continued)**

| Field       | Description                                                                                                                                                                              |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 5<br>HRS26  | Hardware Request Status Channel 26<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 6<br>HRS25  | Hardware Request Status Channel 25<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 7<br>HRS24  | Hardware Request Status Channel 24<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 8<br>HRS23  | Hardware Request Status Channel 23<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 9<br>HRS22  | Hardware Request Status Channel 22<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 10<br>HRS21 | Hardware Request Status Channel 21<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 11<br>HRS20 | Hardware Request Status Channel 20<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 12<br>HRS19 | Hardware Request Status Channel 19<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 13<br>HRS18 | Hardware Request Status Channel 18<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 14<br>HRS17 | Hardware Request Status Channel 17<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 15<br>HRS16 | Hardware Request Status Channel 16<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 16<br>HRS15 | Hardware Request Status Channel 15<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |

*Table continues on the next page...*

**DMA\_HRSL field descriptions (continued)**

| <b>Field</b> | <b>Description</b>                                                                                                                                                                       |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 17<br>HRS14  | Hardware Request Status Channel 14<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 18<br>HRS13  | Hardware Request Status Channel 13<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 19<br>HRS12  | Hardware Request Status Channel 12<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 20<br>HRS11  | Hardware Request Status Channel 11<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 21<br>HRS10  | Hardware Request Status Channel 10<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 22<br>HRS9   | Hardware Request Status Channel 9<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present  |
| 23<br>HRS8   | Hardware Request Status Channel 8<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present  |
| 24<br>HRS7   | Hardware Request Status Channel 7<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present  |
| 25<br>HRS6   | Hardware Request Status Channel 6<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present  |
| 26<br>HRS5   | Hardware Request Status Channel 5<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present  |
| 27<br>HRS4   | Hardware Request Status Channel 4<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present  |
| 28<br>HRS3   | Hardware Request Status Channel 3<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present  |

*Table continues on the next page...*

**DMA\_HRSL field descriptions (continued)**

| Field      | Description                                                                                                                                                                             |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 29<br>HRS2 | Hardware Request Status Channel 2<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 30<br>HRS1 | Hardware Request Status Channel 1<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |
| 31<br>HRS0 | Hardware Request Status Channel 0<br>0 A hardware service request for the corresponding channel is not present<br>1 A hardware service request for the corresponding channel is present |

**16.4.25 Global PCU Output Register (DMA\_GPOR<sub>n</sub>)**

The DMAGPORH and DMAGPORL registers provides coherency information to the Peripheral Coherency Unit (PCU). The DMAGPORH and DMAGPORL registers use a bit map for the implemented channels {16,32,64} to enable a global write signal to the PCU on a per channel basis. When the bit in the corresponding channel is set, the eDMA will signal the PCU whenever that channel is performing a write. The DMAGPORH and DMAGPORL registers perform no functions within the eDMA.

**NOTE**

See the PCU chapter for coherency information and global write usage.

Address: 0h base + 38h offset + (4d × i), where i=0d to 1d

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**DMA\_GPOR<sub>n</sub> field descriptions**

| Field        | Description                                                                                            |
|--------------|--------------------------------------------------------------------------------------------------------|
| 0–31<br>GPOR | Global channel write control<br>Setting a bit specifies the corresponding channel's writes are global. |

### 16.4.26 Channel n Priority Register (DMA\_DCHPRIn)

When fixed-priority channel arbitration is enabled ( $CR[ERCA] = 0$ ), the contents of these registers define the unique priorities associated with each channel within a group. The channel priorities are evaluated by numeric value; for example, 0 is the lowest priority, 1 is the next higher priority, then 2, 3, etc. Software must program the channel priorities with unique values; otherwise, a configuration error is reported. The range of the priority value is limited to the values of 0 through 15. When read, the GRPPRI bits of the DCHPRIn register reflect the current priority level of the group of channels in which the corresponding channel resides. GRPPRI bits are not affected by writes to the DCHPRIn registers. The group priority is assigned in the DMA control register.

Address:  $0h \text{ base} + 100h \text{ offset} + (1d \times i)$ , where  $i=0d$  to  $63d$

| Bit   | 0   | 1   | 2      | 3 | 4     | 5 | 6 | 7 |
|-------|-----|-----|--------|---|-------|---|---|---|
| Read  | ECP | DPA | GRPPRI |   | CHPRI |   |   |   |
| Write |     |     |        |   |       |   |   |   |
| Reset | 0   | 0   | *      | * | *     | * | * | * |

\* Notes:

- CHPRI field: See bit field description.
- GRPPRI field: See bit field description.

#### DMA\_DCHPRIn field descriptions

| Field         | Description                                                                                                                                                                                                                                                                                                                                                                                                             |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>ECP      | Enable Channel Preemption. This field resets to 0.<br>0 Channel n cannot be suspended by a higher priority channel's service request.<br>1 Channel n can be temporarily suspended by the service request of a higher priority channel.                                                                                                                                                                                  |
| 1<br>DPA      | Disable Preempt Ability. This field resets to 0.<br>0 Channel n can suspend a lower priority channel.<br>1 Channel n cannot suspend any channel, regardless of channel priority.                                                                                                                                                                                                                                        |
| 2–3<br>GRPPRI | Channel n Current Group Priority<br>Group priority assigned to this channel group when fixed-priority arbitration is enabled. This field is read-only; writes are ignored.<br><b>NOTE:</b> Reset value for the group and channel priority fields, GRPPRI and CHPRI, is equal to the corresponding channel number for each priority register, that is, $DCHPRI31[GRPPRI] = 0b01$ and $DCHPRI31[CHPRI]$ equals $0b1111$ . |
| 4–7<br>CHPRI  | Channel n Arbitration Priority<br>Channel priority when fixed-priority arbitration is enabled                                                                                                                                                                                                                                                                                                                           |

*Table continues on the next page...*

DMA\_DCHPRI<sub>n</sub> field descriptions (continued)

| Field | Description                                                                                                                                                                                                                    |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | <b>NOTE:</b> Reset value for the group and channel priority fields, GRPPRI and CHPRI, is equal to the corresponding channel number for each priority register, that is, DCHPRI31[GRPPRI] = 0b01 and DCHPRI31[CHPRI] = 0b01111. |

16.4.27 TCD Source Address (DMA\_TCD<sub>n</sub>\_SADDR)

Address: 0h base + 1000h offset + (32d × i), where i=0d to 63d

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

DMA\_TCD<sub>n</sub>\_SADDR field descriptions

| Field         | Description                                                   |
|---------------|---------------------------------------------------------------|
| 0–31<br>SADDR | Source Address<br>Memory address pointing to the source data. |

16.4.28 TCD Transfer Attributes (DMA\_TCD<sub>n</sub>\_ATTR)

Address: 0h base + 1004h offset + (32d × i), where i=0d to 63d

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| Read  |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Write |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

DMA\_TCD<sub>n</sub>\_ATTR field descriptions

| Field        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–4<br>SMOD  | Source Address Modulo<br><br>0 Source address modulo feature is disabled<br>≠0 This value defines a specific address range specified to be the value after SADDR + SOFF calculation is performed on the original register value. Setting this field provides the ability to implement a circular data queue easily. For data queues requiring power-of-2 size bytes, the queue should start at a 0-modulo-size address and the SMOD field should be set to the appropriate value for the queue, freezing the desired number of upper address bits. The value programmed into this field specifies the number of lower address bits allowed to change. For a circular queue application, the SOFF is typically set to the transfer size to implement post-increment addressing with the SMOD function constraining the addresses to a 0-modulo-size range. |
| 5–7<br>SSIZE | Source data transfer size<br><br><b>NOTE:</b> Using a Reserved value causes a configuration error.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |

Table continues on the next page...

**DMA\_TCDn\_ATTR field descriptions (continued)**

| Field          | Description                                                       |
|----------------|-------------------------------------------------------------------|
|                | The eDMA defaults to privileged data access for all transactions. |
| 000            | 8-bit                                                             |
| 001            | 16-bit                                                            |
| 010            | 32-bit                                                            |
| 011            | 64-bit                                                            |
| 100            | Reserved                                                          |
| 101            | 32-byte burst (4 beats of 64 bits)                                |
| 110            | Reserved                                                          |
| 111            | Reserved                                                          |
| 8–12<br>DMOD   | Destination Address Modulo<br><br>See the SMOD definition         |
| 13–15<br>DSIZE | Destination data transfer size<br><br>See the SSIZE definition    |

**16.4.29 TCD Signed Source Address Offset (DMA\_TCDn\_SOFF)**

Address: 0h base + 1006h offset + (32d × i), where i=0d to 63d

|       |      |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| Read  | SOFF |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Write |      |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0    | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

**DMA\_TCDn\_SOFF field descriptions**

| Field        | Description                                                                                                                                                   |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>SOFF | Source address signed offset<br><br>Sign-extended offset applied to the current source address to form the next-state value as each source read is completed. |

**16.4.30 TCD Minor Byte Count (Minor Loop Mapping Disabled) (DMA\_TCDn\_NBYTES\_MLNO)**

This register, or one of the next two registers (TCD\_NBYTES\_MLOFFNO, TCD\_NBYTES\_MLOFFYES), defines the number of bytes to transfer per request. Which register to use depends on whether minor loop mapping is disabled, enabled but not used for this channel, or enabled and used.

TCD word 2 is defined as follows if:

- Minor loop mapping is disabled (CR[EMLM] = 0)

If minor loop mapping is enabled, see the TCD\_NBYTES\_MLOFFNO and TCD\_NBYTES\_MLOFFYES register descriptions for the definition of TCD word 2.

Address: 0h base + 1008h offset + (32d × i), where i=0d to 63d

|       |        |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|--------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0      | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | NBYTES |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |        |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0      | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

### DMA\_TCDn\_NBYTES\_MLNO field descriptions

| Field          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>NBYTES | <p>Minor Byte Transfer Count</p> <p>Number of bytes to be transferred in each service request of the channel. As a channel activates, the appropriate TCD contents load into the eDMA engine, and the appropriate reads and writes perform until the minor byte transfer count has transferred. This is an indivisible operation and cannot be halted. It can, however, be stalled by using the bandwidth control field, or via preemption. After the minor count is exhausted, the SADDR and DADDR values are written back into the TCD memory, the major iteration count is decremented and restored to the TCD memory. If the major iteration count is completed, additional processing is performed.</p> <p><b>NOTE:</b> An NBYTES value of 0x0000_0000 is interpreted as a 4 GB transfer.</p> |

### 16.4.31 TCD Signed Minor Loop Offset (Minor Loop Mapping Enabled and Offset Disabled) (DMA\_TCDn\_NBYTES\_MLOFFNO)

One of three registers (this register, TCD\_NBYTES\_MLNO, or TCD\_NBYTES\_MLOFFYES), defines the number of bytes to transfer per request. Which register to use depends on whether minor loop mapping is disabled, enabled but not used for this channel, or enabled and used.

TCD word 2 is defined as follows if:

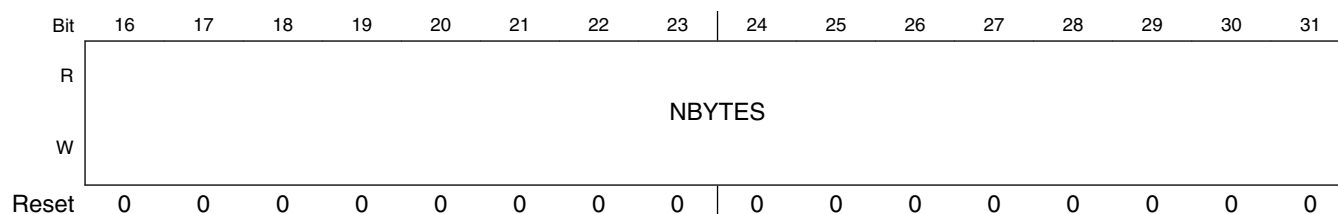
- Minor loop mapping is enabled (CR[EMLM] = 1) and
- SMLOE = 0 and DMLOE = 0

If minor loop mapping is enabled and SMLOE or DMLOE is set, then refer to the TCD\_NBYTES\_MLOFFYES register description. If minor loop mapping is disabled, then refer to the TCD\_NBYTES\_MLNO register description.

Address: 0h base + 1008h offset + (32d × i), where i=0d to 63d

|       |       |       |        |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|-------|-------|--------|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0     | 1     | 2      | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | SMLOE | DMLOE | NBYTES |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |       |       |        |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0     | 0     | 0      | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |



**DMA\_TCDn\_NBYTES\_MLOFFNO field descriptions**

| Field          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>SMLOE     | Source Minor Loop Offset Enable<br><br>Selects whether the minor loop offset is applied to the source address upon minor loop completion.<br><br>0 The minor loop offset is not applied to the SADDR<br>1 The minor loop offset is applied to the SADDR                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 1<br>DMLOE     | Destination Minor Loop Offset enable<br><br>Selects whether the minor loop offset is applied to the destination address upon minor loop completion.<br><br>0 The minor loop offset is not applied to the DADDR<br>1 The minor loop offset is applied to the DADDR                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 2–31<br>NBYTES | Minor Byte Transfer Count<br><br>Number of bytes to be transferred in each service request of the channel.<br><br>As a channel activates, the appropriate TCD contents load into the eDMA engine, and the appropriate reads and writes perform until the minor byte transfer count has transferred. This is an indivisible operation and cannot be halted. It can, however, be stalled by using the bandwidth control field, or via preemption. After the minor count is exhausted, the SADDR and DADDR values are written back into the TCD memory, the major iteration count is decremented and restored to the TCD memory. If the major iteration count is completed, additional processing is performed. |

### 16.4.32 TCD Signed Minor Loop Offset (Minor Loop Mapping and Offset Enabled) (DMA\_TCDn\_NBYTES\_MLOFFYES)

One of three registers (this register, TCD\_NBYTES\_MLNO, or TCD\_NBYTES\_MLOFFNO), defines the number of bytes to transfer per request. Which register to use depends on whether minor loop mapping is disabled, enabled but not used for this channel, or enabled and used.

TCD word 2 is defined as follows if:

- Minor loop mapping is enabled (CR[EMLM] = 1) and
- Minor loop offset is enabled (SMLOE or DMLOE = 1)

If minor loop mapping is enabled and SMLOE and DMLOE are cleared, then refer to the TCD\_NBYTES\_MLOFFNO register description. If minor loop mapping is disabled, then refer to the TCD\_NBYTES\_MLNO register description.

## Memory map/register definition

Address: 0h base + 1008h offset + (32d × i), where i=0d to 63d

|       |       |       |       |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|-------|-------|-------|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0     | 1     | 2     | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | SMLOE | DMLOE | MLOFF |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |       |       |       |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0     | 0     | 0     | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |       |    |    |    |    |    |    |    |        |    |    |    |    |    |    |    |
|-------|-------|----|----|----|----|----|----|----|--------|----|----|----|----|----|----|----|
| Bit   | 16    | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24     | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | MLOFF |    |    |    |    |    |    |    | NBYTES |    |    |    |    |    |    |    |
| W     |       |    |    |    |    |    |    |    |        |    |    |    |    |    |    |    |
| Reset | 0     | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0      | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

### DMA\_TCDn\_NBYTES\_MLOFFYES field descriptions

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>SMLOE      | Source Minor Loop Offset Enable<br><br>Selects whether the minor loop offset is applied to the source address upon minor loop completion.<br><br>0 The minor loop offset is not applied to the SADDR<br>1 The minor loop offset is applied to the SADDR                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 1<br>DMLOE      | Destination Minor Loop Offset enable<br><br>Selects whether the minor loop offset is applied to the destination address upon minor loop completion.<br><br>0 The minor loop offset is not applied to the DADDR<br>1 The minor loop offset is applied to the DADDR                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 2–21<br>MLOFF   | If SMLOE or DMLOE is set, this field represents a sign-extended offset applied to the source or destination address to form the next-state value after the minor loop completes.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 22–31<br>NBYTES | Minor Byte Transfer Count<br><br>Number of bytes to be transferred in each service request of the channel.<br><br>As a channel activates, the appropriate TCD contents load into the eDMA engine, and the appropriate reads and writes perform until the minor byte transfer count has transferred. This is an indivisible operation and cannot be halted. It can, however, be stalled by using the bandwidth control field, or via preemption. After the minor count is exhausted, the SADDR and DADDR values are written back into the TCD memory, the major iteration count is decremented and restored to the TCD memory. If the major iteration count is completed, additional processing is performed. |

## 16.4.33 TCD Last Source Address Adjustment (DMA\_TCDn\_SLAST)

Address: 0h base + 100Ch offset + (32d × i), where i=0d to 63d

|       |       |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0     | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | SLAST |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |       |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0     | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**DMA\_TCDn\_SLAST field descriptions**

| Field         | Description                                                                                                                                                                                                                                                                                                                                                             |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>SLAST | <p>Last Source Address Adjustment</p> <p>Adjustment value added to the source address at the completion of the major iteration count. This value can be applied to restore the source address to the initial value, or adjust the address to reference the next data structure.</p> <p>This register uses two's complement notation; the overflow bit is discarded.</p> |

**16.4.34 TCD Destination Address (DMA\_TCDn\_DADDR)**

Address: 0h base + 1010h offset + (32d × i), where i=0d to 63d

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**DMA\_TCDn\_DADDR field descriptions**

| Field         | Description                                                                        |
|---------------|------------------------------------------------------------------------------------|
| 0–31<br>DADDR | <p>Destination Address</p> <p>Memory address pointing to the destination data.</p> |

**16.4.35 TCD Current Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA\_TCDn\_CITER\_ELINKYES)**

If TCDn\_CITER[ELINK] is set, the TCDn\_CITER register is defined as follows.

Address: 0h base + 1014h offset + (32d × i), where i=0d to 63d

|       |   |   |   |   |   |   |   |   |
|-------|---|---|---|---|---|---|---|---|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Read  |   |   |   |   |   |   |   |   |
| Write |   |   |   |   |   |   |   |   |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|       |   |   |    |    |    |    |    |    |
|-------|---|---|----|----|----|----|----|----|
| Bit   | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| Read  |   |   |    |    |    |    |    |    |
| Write |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

**DMA\_TCDn\_CITER\_ELINKYES field descriptions**

| Field      | Description                                              |
|------------|----------------------------------------------------------|
| 0<br>ELINK | Enable channel-to-channel linking on minor-loop complete |

*Table continues on the next page...*

**DMA\_TCDn\_CITER\_ELINKYES field descriptions (continued)**

| Field         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               | <p>As the channel completes the minor loop, this flag enables linking to another channel, defined by the LINKCH field. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] bit of the specified channel.</p> <p>If channel linking is disabled, the CITER value is extended to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of the MAJORELINK channel linking.</p> <p><b>NOTE:</b> This bit must be equal to the BITER[ELINK] bit; otherwise, a configuration error is reported.</p> <p>0 The channel-to-channel linking is disabled<br/>1 The channel-to-channel linking is enabled</p>                                                                                                                                                   |
| 1–6<br>LINKCH | <p>Minor Loop Link Channel Number</p> <p>If channel-to-channel linking is enabled (ELINK = 1), then after the minor loop is exhausted, the eDMA engine initiates a channel service request to the channel defined by this field by setting that channel's TCDn_CSR[START] bit.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 7–15<br>CITER | <p>Current Major Iteration Count</p> <p>This 9-bit (ELINK = 1) or 15-bit (ELINK = 0) count represents the current major loop count for the channel. It is decremented each time the minor loop is completed and updated in the transfer control descriptor memory. After the major iteration count is exhausted, the channel performs a number of operations, for example, final source and destination address calculations, optionally generating an interrupt to signal channel completion before reloading the CITER field from the Beginning Iteration Count (BITER) field.</p> <p><b>NOTE:</b> When the CITER field is initially loaded by software, it must be set to the same value as that contained in the BITER field.</p> <p><b>NOTE:</b> If the channel is configured to execute a single service request, the initial values of BITER and CITER should be 0x0001.</p> |

**16.4.36 TCD Current Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA\_TCDn\_CITER\_ELINKNO)**

If TCDn\_CITER[ELINK] is cleared, the TCDn\_CITER register is defined as follows.

Address: 0h base + 1014h offset + (32d × i), where i=0d to 63d

|       |       |       |    |    |    |    |    |    |
|-------|-------|-------|----|----|----|----|----|----|
| Bit   | 0     | 1     | 2  | 3  | 4  | 5  | 6  | 7  |
| Read  | ELINK | CITER |    |    |    |    |    |    |
| Write |       |       |    |    |    |    |    |    |
| Reset | 0     | 0     | 0  | 0  | 0  | 0  | 0  | 0  |
| Bit   | 8     | 9     | 10 | 11 | 12 | 13 | 14 | 15 |
| Read  | CITER |       |    |    |    |    |    |    |
| Write |       |       |    |    |    |    |    |    |
| Reset | 0     | 0     | 0  | 0  | 0  | 0  | 0  | 0  |

**DMA\_TCDn\_CITER\_ELINKNO field descriptions**

| Field         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>ELINK    | <p>Enable channel-to-channel linking on minor-loop complete</p> <p>As the channel completes the minor loop, this flag enables linking to another channel, defined by the LINKCH field. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] bit of the specified channel.</p> <p>If channel linking is disabled, the CITER value is extended to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of the MAJORELINK channel linking.</p> <p><b>NOTE:</b> This bit must be equal to the BITER[ELINK] bit; otherwise, a configuration error is reported.</p> <p>0 The channel-to-channel linking is disabled<br/>1 The channel-to-channel linking is enabled</p>                                                                                   |
| 1–15<br>CITER | <p>Current Major Iteration Count</p> <p>This 9-bit (ELINK = 1) or 15-bit (ELINK = 0) count represents the current major loop count for the channel. It is decremented each time the minor loop is completed and updated in the transfer control descriptor memory. After the major iteration count is exhausted, the channel performs a number of operations, for example, final source and destination address calculations, optionally generating an interrupt to signal channel completion before reloading the CITER field from the Beginning Iteration Count (BITER) field.</p> <p><b>NOTE:</b> When the CITER field is initially loaded by software, it must be set to the same value as that contained in the BITER field.</p> <p><b>NOTE:</b> If the channel is configured to execute a single service request, the initial values of BITER and CITER should be 0x0001.</p> |

**16.4.37 TCD Signed Destination Address Offset (DMA\_TCDn\_DOFF)**

Address: 0h base + 1016h offset + (32d × i), where i=0d to 63d

|       |      |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| Read  | DOFF |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Write |      |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0    | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

**DMA\_TCDn\_DOFF field descriptions**

| Field        | Description                                                                                                                                                                          |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>DOFF | <p>Destination Address Signed Offset</p> <p>Sign-extended offset applied to the current destination address to form the next-state value as each destination write is completed.</p> |

### 16.4.38 TCD Last Destination Address Adjustment/Scatter Gather Address (DMA\_TCDn\_DLASTSGA)

Address: 0h base + 1018h offset + (32d × i), where i=0d to 63d

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

#### DMA\_TCDn\_DLASTSGA field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>DLASTSGA | <p>Destination last address adjustment or the memory address for the next transfer control descriptor to be loaded into this channel (scatter/gather).</p> <p>If (TCDn_CSR[ESG] = 0) then:</p> <ul style="list-style-type: none"> <li>Adjustment value added to the destination address at the completion of the major iteration count. This value can apply to restore the destination address to the initial value or adjust the address to reference the next data structure.</li> <li>This field uses two's complement notation for the final destination address adjustment.</li> </ul> <p>Otherwise:</p> <ul style="list-style-type: none"> <li>This address points to the beginning of a 0-modulo-32-byte region containing the next transfer control descriptor to be loaded into this channel. This channel reload is performed as the major iteration count completes. The scatter/gather address must be 0-modulo-32-byte, otherwise a configuration error is reported.</li> </ul> |

### 16.4.39 TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Enabled) (DMA\_TCDn\_BITER\_ELINKYES)

If the TCDn\_BITER[ELINK] bit is set, the TCDn\_BITER register is defined as follows.

Address: 0h base + 101Ch offset + (32d × i), where i=0d to 63d

|       |   |   |   |   |   |   |   |   |
|-------|---|---|---|---|---|---|---|---|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Read  |   |   |   |   |   |   |   |   |
| Write |   |   |   |   |   |   |   |   |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

|       |   |   |    |    |    |    |    |    |
|-------|---|---|----|----|----|----|----|----|
| Bit   | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| Read  |   |   |    |    |    |    |    |    |
| Write |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

#### DMA\_TCDn\_BITER\_ELINKYES field descriptions

| Field      | Description                                                                                                                                                                                                                                                                             |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>ELINK | <p>Enables channel-to-channel linking on minor loop complete</p> <p>As the channel completes the minor loop, this flag enables the linking to another channel, defined by BITER[LINKCH]. The link target channel initiates a channel service request via an internal mechanism that</p> |

*Table continues on the next page...*

**DMA\_TCDn\_BITER\_ELINKYES field descriptions (continued)**

| Field         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               | <p>sets the TCDn_CSR[START] bit of the specified channel. If channel linking disables, the BITER value extends to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of the MAJORELINK channel linking.</p> <p><b>NOTE:</b> When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, the contents of this field are reloaded into the CITER field.</p> <p>0 The channel-to-channel linking is disabled<br/>1 The channel-to-channel linking is enabled</p>                                                                |
| 1–6<br>LINKCH | <p>Link Channel Number</p> <p>If channel-to-channel linking is enabled (ELINK = 1), then after the minor loop is exhausted, the eDMA engine initiates a channel service request at the channel defined by this field by setting that channel's TCDn_CSR[START] bit.</p> <p><b>NOTE:</b> When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, the contents of this field are reloaded into the CITER field.</p>                                                                                                                                                                           |
| 7–15<br>BITER | <p>Starting major iteration count</p> <p>As the transfer control descriptor is first loaded by software, this 9-bit (ELINK = 1) or 15-bit (ELINK = 0) field must be equal to the value in the CITER field. As the major iteration count is exhausted, the contents of this field are reloaded into the CITER field.</p> <p><b>NOTE:</b> When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, the contents of this field are reloaded into the CITER field. If the channel is configured to execute a single service request, the initial values of BITER and CITER should be 0x0001.</p> |

**16.4.40 TCD Beginning Minor Loop Link, Major Loop Count (Channel Linking Disabled) (DMA\_TCDn\_BITER\_ELINKNO)**

If the TCDn\_BITER[ELINK] bit is cleared, the TCDn\_BITER register is defined as follows.

Address: 0h base + 101Ch offset + (32d × i), where i=0d to 63d

|       |       |   |    |    |       |    |    |    |
|-------|-------|---|----|----|-------|----|----|----|
| Bit   | 0     | 1 | 2  | 3  | 4     | 5  | 6  | 7  |
| Read  | ELINK |   |    |    | BITER |    |    |    |
| Write |       |   |    |    |       |    |    |    |
| Reset | 0     | 0 | 0  | 0  | 0     | 0  | 0  | 0  |
| Bit   | 8     | 9 | 10 | 11 | 12    | 13 | 14 | 15 |
| Read  | BITER |   |    |    |       |    |    |    |
| Write |       |   |    |    |       |    |    |    |
| Reset | 0     | 0 | 0  | 0  | 0     | 0  | 0  | 0  |

## DMA\_TCDn\_BITER\_ELINKNO field descriptions

| Field         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>ELINK    | <p>Enables channel-to-channel linking on minor loop complete</p> <p>As the channel completes the minor loop, this flag enables the linking to another channel, defined by BITER[LINKCH]. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] bit of the specified channel. If channel linking is disabled, the BITER value extends to 15 bits in place of a link channel number. If the major loop is exhausted, this link mechanism is suppressed in favor of the MAJORELINK channel linking.</p> <p><b>NOTE:</b> When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, the contents of this field are reloaded into the CITER field.</p> <p>0 The channel-to-channel linking is disabled<br/>1 The channel-to-channel linking is enabled</p> |
| 1–15<br>BITER | <p>Starting Major Iteration Count</p> <p>As the transfer control descriptor is first loaded by software, this 9-bit (ELINK = 1) or 15-bit (ELINK = 0) field must be equal to the value in the CITER field. As the major iteration count is exhausted, the contents of this field are reloaded into the CITER field.</p> <p><b>NOTE:</b> When the software loads the TCD, this field must be set equal to the corresponding CITER field; otherwise, a configuration error is reported. As the major iteration count is exhausted, the contents of this field is reloaded into the CITER field. If the channel is configured to execute a single service request, the initial values of BITER and CITER should be 0x0001.</p>                                                                                                                                                                                                                       |

## 16.4.41 TCD Control and Status (DMA\_TCDn\_CSR)

Address: 0h base + 101Eh offset + (32d × i), where i=0d to 63d

|       |      |        |            |             |      |         |          |       |
|-------|------|--------|------------|-------------|------|---------|----------|-------|
| Bit   | 0    | 1      | 2          | 3           | 4    | 5       | 6        | 7     |
| Read  | BWC  |        |            | MAJORLINKCH |      |         |          |       |
| Write |      |        |            |             |      |         |          |       |
| Reset | 0    | 0      | 0          | 0           | 0    | 0       | 0        | 0     |
| Bit   | 8    | 9      | 10         | 11          | 12   | 13      | 14       | 15    |
| Read  | DONE | ACTIVE | MAJORELINK | ESG         | DREQ | INTHALF | INTMAJOR | START |
| Write |      |        |            |             |      |         |          |       |
| Reset | 0    | 0      | 0          | 0           | 0    | 0       | 0        | 0     |

## DMA\_TCDn\_CSR field descriptions

| Field      | Description                                                                                                                                                                                                                                                                                                                                                                          |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–1<br>BWC | <p>Bandwidth Control</p> <p>Throttles the amount of bus bandwidth consumed by the eDMA. Generally, as the eDMA processes the minor loop, it continuously generates read/write sequences until the minor count is exhausted. This field forces the eDMA to stall after the completion of each read/write access to control the bus request bandwidth seen by the crossbar switch.</p> |

Table continues on the next page...



**DMA\_TCDn\_CSR field descriptions (continued)**

| Field              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                    | <p><b>NOTE:</b> If the source and destination sizes are equal, this field is ignored between the first and second transfers and after the last write of each minor loop. This behavior is a side effect of reducing start-up latency.</p> <p>00 No eDMA engine stalls.</p> <p>01 Enable eDMA master high-priority elevation (HPE) mode. No eDMA engine stalls.</p> <p>10 eDMA engine stalls for 4 cycles after each R/W.</p> <p>11 eDMA engine stalls for 8 cycles after each R/W.</p>                                                                                                                                                                                                                                                                                                    |
| 2–7<br>MAJORLINKCH | <p>Major Loop Link Channel Number</p> <p>If (MAJORELINK = 0) then:</p> <ul style="list-style-type: none"> <li>No channel-to-channel linking, or chaining, is performed after the major loop counter is exhausted.</li> </ul> <p>Otherwise:</p> <ul style="list-style-type: none"> <li>After the major loop counter is exhausted, the eDMA engine initiates a channel service request at the channel defined by this field by setting that channel's TCDn_CSR[START] bit.</li> </ul>                                                                                                                                                                                                                                                                                                       |
| 8<br>DONE          | <p>Channel Done</p> <p>This flag indicates the eDMA has completed the major loop. The eDMA engine sets it as the CITER count reaches zero. The software clears it, or the hardware when the channel is activated.</p> <p>The access of this field is W0C, write-zero-to-clear.</p> <p><b>NOTE:</b> This bit must be cleared to write the MAJORELINK or ESG bits.</p>                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 9<br>ACTIVE        | <p>Channel Active</p> <p>This flag signals the channel is currently in execution. It is set when channel service begins, and is cleared by the eDMA as the minor loop completes or when any error condition is detected.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 10<br>MAJORELINK   | <p>Enable channel-to-channel linking on major loop complete</p> <p>As the channel completes the major loop, this flag enables the linking to another channel, defined by MAJORLINKCH. The link target channel initiates a channel service request via an internal mechanism that sets the TCDn_CSR[START] bit of the specified channel.</p> <p><b>NOTE:</b> To support the dynamic linking coherency model, this field is forced to zero when written to while the TCDn_CSR[DONE] bit is set.</p> <p>0 The channel-to-channel linking is disabled.</p> <p>1 The channel-to-channel linking is enabled.</p>                                                                                                                                                                                |
| 11<br>ESG          | <p>Enable Scatter/Gather Processing</p> <p>As the channel completes the major loop, this flag enables scatter/gather processing in the current channel. If enabled, the eDMA engine uses DLASTSGA as a memory pointer to a 0-modulo-32 address containing a 32-byte data structure loaded as the transfer control descriptor into the local memory.</p> <p><b>NOTE:</b> To support the dynamic scatter/gather coherency model, this field is forced to zero when written to while the TCDn_CSR[DONE] bit is set.</p> <p>0 The current channel's TCD is normal format.</p> <p>1 The current channel's TCD specifies a scatter gather format. The DLASTSGA field provides a memory pointer to the next TCD to be loaded into this channel after the major loop completes its execution.</p> |
| 12<br>DREQ         | <p>Disable Request</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

*Table continues on the next page...*

## DMA\_TCDn\_CSR field descriptions (continued)

| Field          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | <p>If this flag is set, the eDMA hardware automatically clears the corresponding ERQ {H,L} bit when the current major iteration count reaches zero.</p> <p>0 The channel's ERQ {H,L} bit is not affected.<br/> 1 The channel's ERQ {H,L} bit is cleared when the major loop is complete.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 13<br>INTHALF  | <p>Enable an interrupt when major counter is half complete.</p> <p>If this flag is set, the channel generates an interrupt request by setting the appropriate bit in the INT {H,L} register when the current major iteration count reaches the halfway point. Specifically, the comparison performed by the eDMA engine is (CITER == (BITER &gt;&gt; 1)). This halfway point interrupt request is provided to support double-buffered, also known as ping-pong, schemes or other types of data movement where the processor needs an early indication of the transfer's progress.</p> <p><b>NOTE:</b> If BITER = 1, do not use INTHALF. Use INTMAJOR instead.</p> <p>0 The half-point interrupt is disabled.<br/> 1 The half-point interrupt is enabled.</p> |
| 14<br>INTMAJOR | <p>Enable an interrupt when major iteration count completes.</p> <p>If this flag is set, the channel generates an interrupt request by setting the appropriate bit in the INT when the current major iteration count reaches zero.</p> <p>0 The end-of-major loop interrupt is disabled.<br/> 1 The end-of-major loop interrupt is enabled.</p>                                                                                                                                                                                                                                                                                                                                                                                                              |
| 15<br>START    | <p>Channel Start</p> <p>If this flag is set, the channel is requesting service. The eDMA hardware automatically clears this flag after the channel begins execution.</p> <p>0 The channel is not explicitly started.<br/> 1 The channel is explicitly started via a software initiated service request.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

## 16.5 Functional description

The operation of the eDMA is described in the following subsections.

### 16.5.1 eDMA basic data flow

The basic flow of a data transfer can be partitioned into three segments.

As shown in the following diagram, the first segment involves the channel activation:

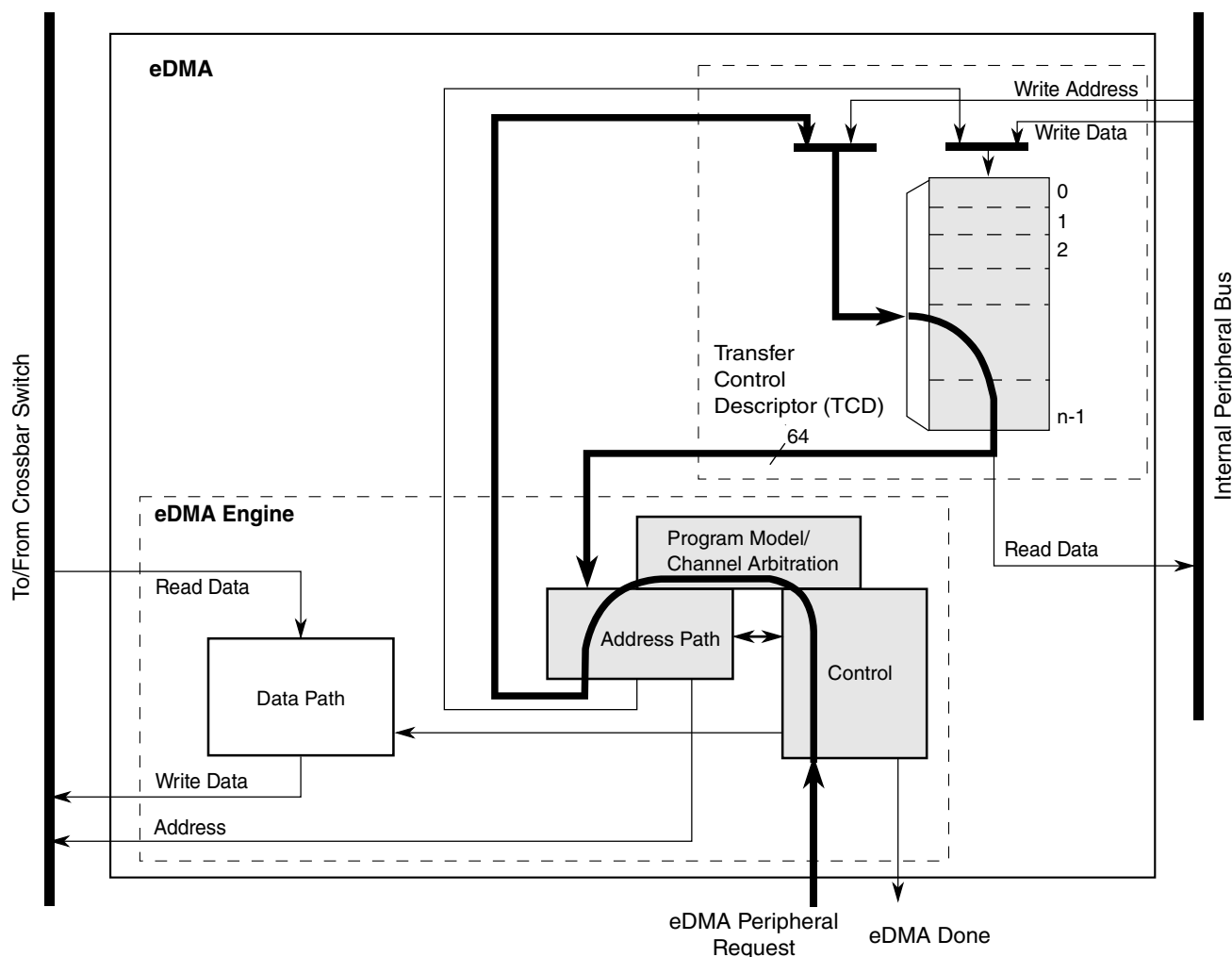


**Figure 16-1065. eDMA operation, part 1**

This example uses the assertion of the eDMA peripheral request signal to request service for channel  $n$ . Channel activation via software and the  $\text{TCD}_n\text{-CSR}[\text{START}]$  bit follows the same basic flow as peripheral requests. The eDMA request input signal is registered internally and then routed through the eDMA engine: first through the control module, then into the program model and channel arbitration. In the next cycle, the channel arbitration performs, using the fixed-priority or round-robin algorithm. After arbitration is complete, the activated channel number is sent through the address path and converted into the required address to access the local memory for  $\text{TCD}_n$ . Next, the TCD memory is accessed and the required descriptor read from the local memory and loaded into the eDMA engine address path channel  $x$  or  $y$  registers. The TCD memory is 64 bits wide to minimize the time needed to fetch the activated channel descriptor and load it into the address path channel  $x$  or  $y$  registers.

The following diagram illustrates the second part of the basic data flow:

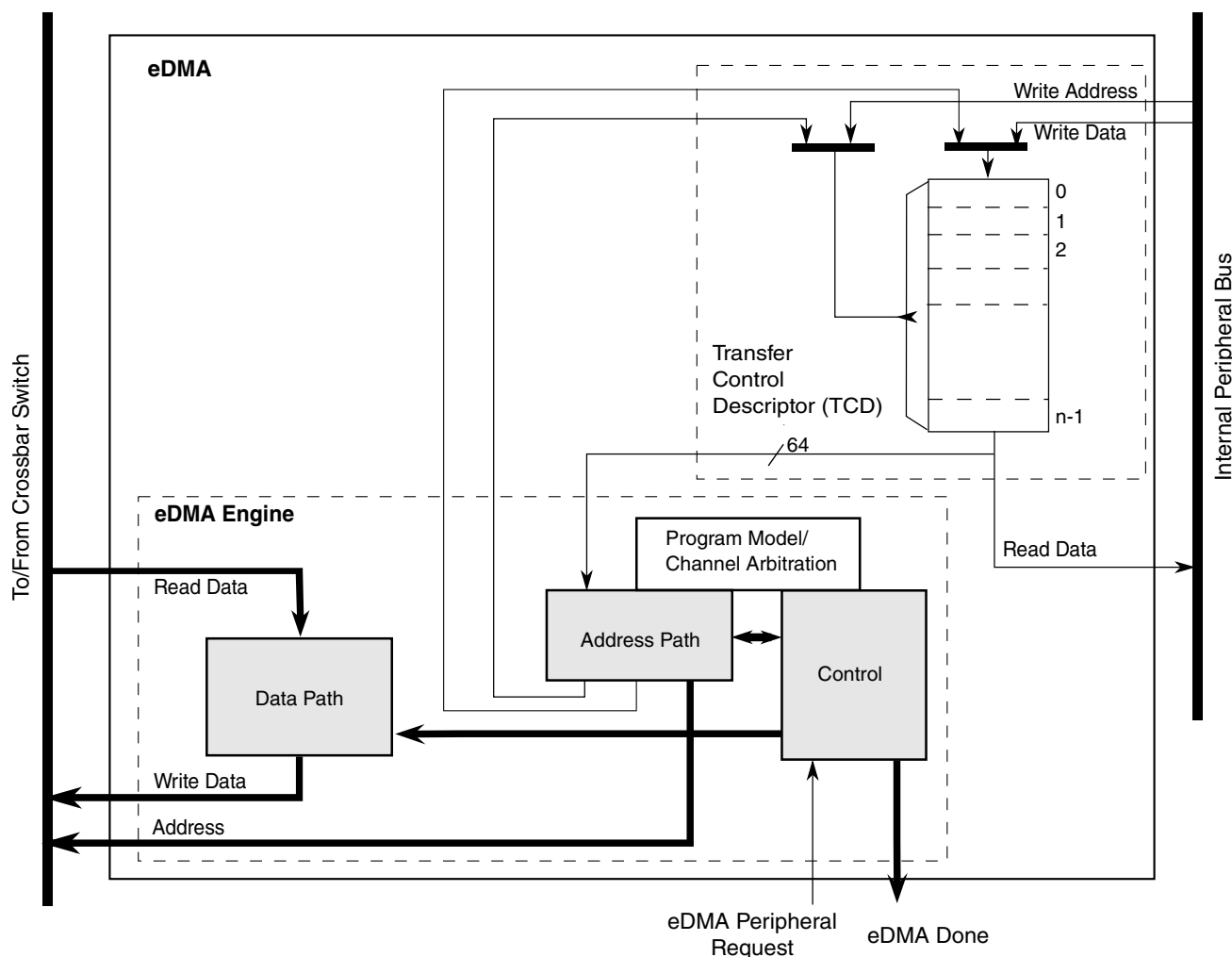


**Figure 16-1066. eDMA operation, part 2**

The modules associated with the data transfer (address path, data path, and control) sequence through the required source reads and destination writes to perform the actual data movement. The source reads are initiated and the fetched data is temporarily stored in the data path block until it is gated onto the internal bus during the destination write. This source read/destination write processing continues until the minor byte count has transferred.

After the minor byte count has moved, the final phase of the basic data flow is performed. In this segment, the address path logic performs the required updates to certain fields in the appropriate TCD, for example, SADDR, DADDR, CITER. If the major iteration count is exhausted, additional operations are performed. These include the final address adjustments and reloading of the BITER field into the CITER. Assertion of an optional interrupt request also occurs at this time, as does a possible fetch of a new TCD from memory using the scatter/gather address pointer included in the descriptor (if scatter/gather is enabled). The updates to the TCD memory and the assertion of an interrupt request are shown in the following diagram.

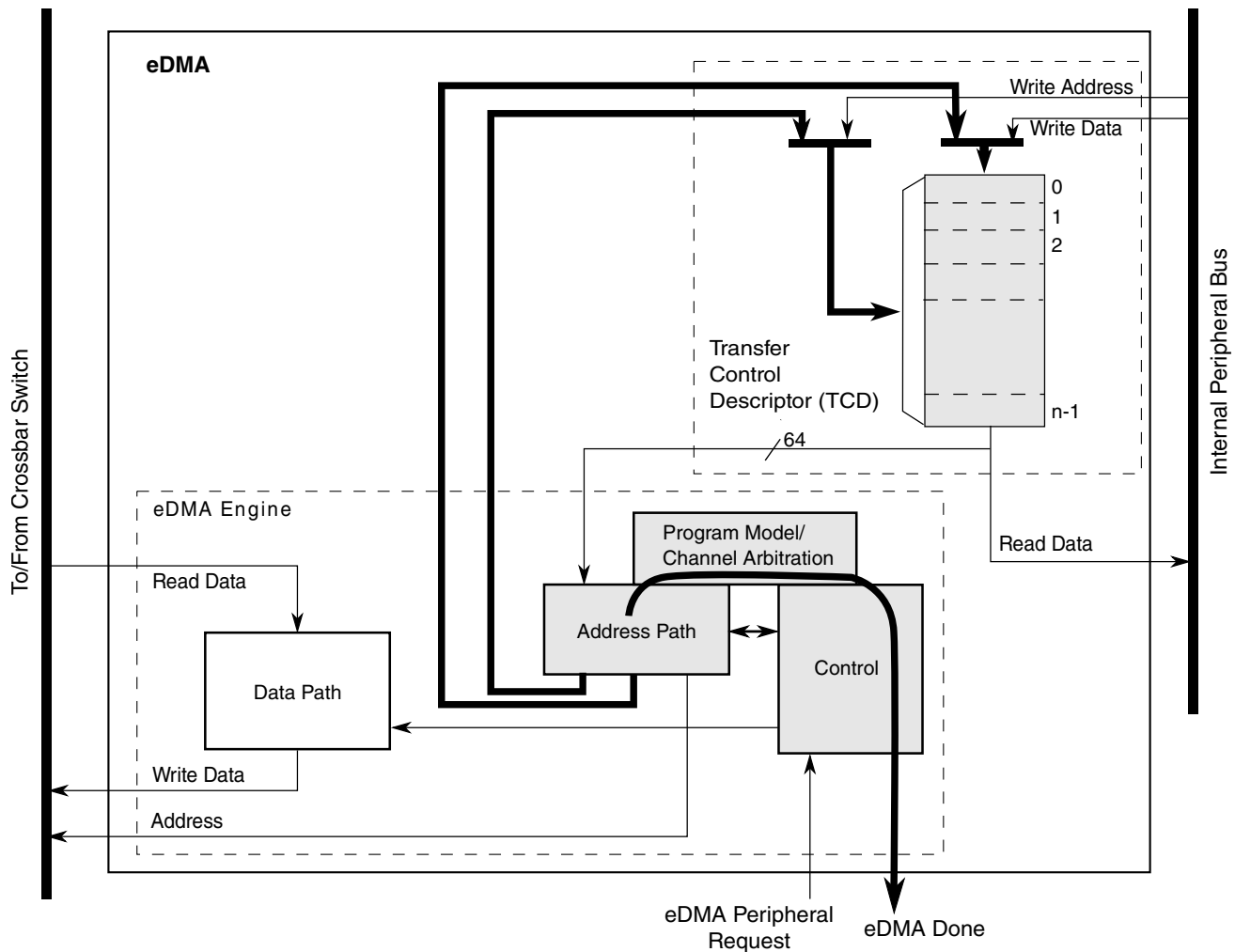


Figure 16-1067. eDMA operation, part 3

### 16.5.2 Fault reporting and handling

Channel errors are reported in the Error Status register (DMAx\_ES) and can be caused by:

- A configuration error, which is an illegal setting in the transfer-control descriptor or an illegal priority register setting in Fixed-Arbitration mode
- An uncorrectable ECC error, or
- An error termination to a bus master read or write cycle

A configuration error is reported when the starting source or destination address, source or destination offsets, minor loop byte count, or the transfer size represent an inconsistent state. Each of these possible causes are detailed below:

- The addresses and offsets must be aligned on 0-modulo-transfer-size boundaries.

- The minor loop byte count must be a multiple of the source and destination transfer sizes.
- All source reads and destination writes must be configured to the natural boundary of the programmed transfer size respectively.
- In fixed arbitration mode, a configuration error is caused by any two channel priorities being equal within a group, or any group priority levels being equal among the groups. All channel priority levels within a group must be unique and all group priority levels among the groups must be unique when fixed arbitration mode is enabled.

### **NOTE**

When two channels have the same priority, a channel priority error exists and will be reported in the Error Status register. However, the channel number will not be reported in the Error Status register. When all of the channel priorities within a group are not unique, the channel number selected by arbitration is undetermined.

To aid in Channel Priority Error (CPE) debug, set the Halt On Error bit in the DMA's Control Register. If all of the channel priorities within a group are not unique, the DMA will be halted after the CPE error is recorded. The DMA will remain halted and will not process any channel service requests. Once all of the channel priorities are set to unique numbers, the DMA may be enabled again by clearing the Halt bit.

- If a scatter/gather operation is enabled upon channel completion, a configuration error is reported if the scatter/gather address (DLAST\_SGA) is not aligned on a 32-byte boundary.
- If minor loop channel linking is enabled upon channel completion, a configuration error is reported when the link is attempted if the TCDn\_CITER[E\_LINK] bit does not equal the TCDn\_BITER[E\_LINK] bit.

If enabled, all configuration error conditions, except the scatter/gather and minor-loop link errors, report as the channel activates and asserts an error interrupt request. A scatter/gather configuration error is reported when the scatter/gather operation begins at major loop completion when properly enabled. A minor loop channel link configuration error is reported when the link operation is serviced at minor loop completion.

If a system bus read or write is terminated with an error, the data transfer is stopped and the appropriate bus error flag set. In this case, the state of the channel's transfer control descriptor is updated by the eDMA engine with the current source address, destination address, and current iteration count at the point of the fault. When a system bus error occurs, the channel terminates after the next transfer. Due to pipeline effect, the next transfer is already in progress when the bus error is received by the eDMA. If a bus error occurs on the last read prior to beginning the write sequence, the write executes using the data captured during the bus error. If a bus error occurs on the last write prior to switching to the next read sequence, the read sequence executes before the channel terminates due to the destination bus error.

If an uncorrectable ECC error occurs during a peripheral bus access, the access is terminated with a bus error. The occurrence of an uncorrectable ECC error during an eDMA engine read causes the eDMA engine to stop the active channel immediately. The ECC error and channel number are recorded in the eDMA's Error Status register.

A transfer may be cancelled by software with the CR[CX] bit. When a cancel transfer request is recognized, the DMA engine stops processing the channel. The current read-write sequence is allowed to finish. If the cancel occurs on the last read-write sequence of a major or minor loop, the cancel request is discarded and the channel retires normally.

The error cancel transfer is the same as a cancel transfer except the Error Status register (DMAx\_ES) is updated with the cancelled channel number and ECX is set. The TCD of a cancelled channel contains the source and destination addresses of the last transfer saved in the TCD. If the channel needs to be restarted, you must re-initialize the TCD because the aforementioned fields no longer represent the original parameters. When a transfer is cancelled by the error cancel transfer mechanism, the channel number is loaded into DMA\_ES[ERRCHN] and ECX and VLD are set. In addition, an error interrupt may be generated if enabled.

### NOTE

The cancel transfer request allows the user to stop a large data transfer in the event the full data transfer is no longer needed. The cancel transfer bit does not abort the channel. It simply stops the transferring of data and then retires the channel through its normal shutdown sequence. The application software must handle the context of the cancel. If an interrupt is desired (or not), then the interrupt should be enabled (or disabled) before the cancel request. The application software must clean up the transfer control descriptor since the full transfer did not occur.

The occurrence of any error causes the eDMA engine to stop normal processing of the active channel immediately (it goes to its error processing states and the transaction to the system bus still has pipeline effect), and the appropriate channel bit in the eDMA error register is asserted. At the same time, the details of the error condition are loaded into the Error Status register (DMAx\_ES). The major loop complete indicators, setting the transfer control descriptor DONE flag and the possible assertion of an interrupt request, are not affected when an error is detected. After the error status has been updated, the eDMA engine continues operating by servicing the next appropriate channel. A channel that experiences an error condition is not automatically disabled. If a channel is terminated by an error and then issues another service request before the error is fixed, that channel executes and terminates with the same error condition.

### **16.5.3 Channel preemption**

Channel preemption is enabled on a per-channel basis by setting the DCHPRIn[ECP] bit. Channel preemption allows the executing channel's data transfers to temporarily suspend in favor of starting a higher priority channel. After the preempting channel has completed all its minor loop data transfers, the preempted channel is restored and resumes execution. After the restored channel completes one read/write sequence, it is again eligible for preemption. If any higher priority channel is requesting service, the restored channel is suspended and the higher priority channel is serviced. Nested preemption, that is, attempting to preempt a preempting channel, is not supported. After a preempting channel begins execution, it cannot be preempted. Preemption is available only when fixed arbitration is selected for both group and channel arbitration modes.

A channel's ability to preempt another channel can be disabled by setting DCHPRIn[DPA]. When a channel's preempt ability is disabled, that channel cannot suspend a lower priority channel's data transfer, regardless of the lower priority channel's ECP setting. This allows for a pool of low priority, large data-moving channels to be defined. These low priority channels can be configured to not preempt each other, thus preventing a low priority channel from consuming the preempt slot normally available to a true, high priority channel.

### **16.5.4 Performance**

This section addresses the performance of the eDMA module, focusing on two separate metrics:



- In the traditional data movement context, performance is best expressed as the peak data transfer rates achieved using the eDMA. In most implementations, this transfer rate is limited by the speed of the source and destination address spaces.
- In a second context where device-paced movement of single data values to/from peripherals is dominant, a measure of the requests that can be serviced in a fixed time is a more relevant metric. In this environment, the speed of the source and destination address spaces remains important. However, the microarchitecture of the eDMA also factors significantly into the resulting metric.

### 16.5.4.1 Peak transfer rates

The peak transfer rates for several different source and destination transfers are shown in the following tables. These tables assume:

- Internal SRAM can be accessed with zero wait-states when viewed from the system bus data phase
- All internal peripheral bus reads require two wait-states, and internal peripheral bus writes three wait-states, when viewed from the system bus data phase
- All internal peripheral bus accesses are 32-bits in size

#### NOTE

All architectures will not meet the assumptions listed above.  
See the SRAM configuration section for more information.

This table compares peak transfer rates based on different possible system speeds. Specific chips/devices may not support all system speeds listed.

**Table 16-1068. eDMA peak transfer rates (Mbytes/sec)**

| System Speed, Width | Internal SRAM-to-Internal SRAM | 32 bit internal peripheral bus-to-Internal SRAM | Internal SRAM-to-32 bit internal peripheral bus |
|---------------------|--------------------------------|-------------------------------------------------|-------------------------------------------------|
| 66.7 MHz, 64 bit    | 266.7                          | 66.6                                            | 53.3                                            |
| 83.3 MHz, 64 bit    | 333.3                          | 83.3                                            | 66.7                                            |
| 100.0 MHz, 64 bit   | 400.0                          | 100.0                                           | 80.0                                            |
| 133.3 MHz, 64 bit   | 533.3                          | 133.3                                           | 106.7                                           |
| 150.0 MHz, 64 bit   | 600.0                          | 150.0                                           | 120.0                                           |

Internal-SRAM-to-internal-SRAM transfers occur at the core's datapath width. For all transfers involving the internal peripheral bus, 32-bit transfer sizes are used. In all cases, the transfer rate includes the time to read the source plus the time to write the destination.

### 16.5.4.2 Peak request rates

The second performance metric is a measure of the number of DMA requests that can be serviced in a given amount of time. For this metric, assume that the peripheral request causes the channel to move a single internal peripheral bus-mapped operand to/from internal SRAM. The same timing assumptions used in the previous example apply to this calculation. In particular, this metric also reflects the time required to activate the channel.

The eDMA design supports the following hardware service request sequence. Note that the exact timing from Cycle 7 is a function of the response times for the channel's read and write accesses. In the case of an internal peripheral bus read and internal SRAM write, the combined data phase time is 4 cycles. For an SRAM read and internal peripheral bus write, it is 5 cycles.

**Table 16-1069. Hardware service request process**

| Cycle                                                     |                                                  | Description                                                                                                                                                                                                                                            |
|-----------------------------------------------------------|--------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| With internal peripheral bus read and internal SRAM write | With SRAM read and internal peripheral bus write |                                                                                                                                                                                                                                                        |
| 1                                                         |                                                  | eDMA peripheral request is asserted.                                                                                                                                                                                                                   |
| 2                                                         |                                                  | The eDMA peripheral request is registered locally in the eDMA module and qualified. TCD <sub>n</sub> _CSR[START] bit initiated requests start at this point with the registering of the user write to TCD <sub>n</sub> word 7.                         |
| 3                                                         |                                                  | Channel arbitration begins.                                                                                                                                                                                                                            |
| 4                                                         |                                                  | Channel arbitration completes. The transfer control descriptor local memory read is initiated.                                                                                                                                                         |
| 5–6                                                       |                                                  | The first two parts of the activated channel's TCD is read from the local memory. The memory width to the eDMA engine is 64 bits, so the entire descriptor can be accessed in four cycles                                                              |
| 7                                                         |                                                  | The first system bus read cycle is initiated, as the third part of the channel's TCD is read from the local memory. Depending on the state of the crossbar switch, arbitration at the system bus may insert an additional cycle of delay here.         |
| 8–11                                                      | 8–12                                             | The last part of the TCD is read in. This cycle represents the first data phase for the read, and the address phase for the destination write.                                                                                                         |
| 12                                                        | 13                                               | This cycle represents the data phase of the last destination write.                                                                                                                                                                                    |
| 13                                                        | 14                                               | The eDMA engine completes the execution of the inner minor loop and prepares to write back the required TCD <sub>n</sub> fields into the local memory. The TCD <sub>n</sub> word 7 is read and checked for channel linking or scatter/gather requests. |

*Table continues on the next page...*

**Table 16-1069. Hardware service request process (continued)**

| Cycle                                                     |                                                  | Description                                                                                                                                                                   |
|-----------------------------------------------------------|--------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| With internal peripheral bus read and internal SRAM write | With SRAM read and internal peripheral bus write |                                                                                                                                                                               |
| 14                                                        | 15                                               | The appropriate fields in the first part of the TCD <sub>n</sub> are written back into the local memory.                                                                      |
| 15                                                        | 16                                               | The fields in the second part of the TCD <sub>n</sub> are written back into the local memory. This cycle coincides with the next channel arbitration cycle start.             |
| 16                                                        | 17                                               | The next channel to be activated performs the read of the first part of its TCD from the local memory. This is equivalent to Cycle 4 for the first channel's service request. |

Assuming zero wait states on the system bus, DMA requests can be processed every 9 cycles. Assuming an average of the access times associated with internal peripheral bus-to-SRAM (4 cycles) and SRAM-to-internal peripheral bus (5 cycles), DMA requests can be processed every 11.5 cycles ( $4 + (4+5)/2 + 3$ ). This is the time from Cycle 4 to Cycle  $x + 5$ . The resulting peak request rate, as a function of the system frequency, is shown in the following table.

**Table 16-1070. eDMA peak request rate (MReq/sec)**

| System frequency (MHz) | Request rate<br>with zero wait states | Request rate<br>with wait states |
|------------------------|---------------------------------------|----------------------------------|
| 48.0                   | 5.3                                   | 4.2                              |
| 66.6                   | 7.4                                   | 5.8                              |
| 83.3                   | 9.2                                   | 7.2                              |
| 100.0                  | 11.1                                  | 8.7                              |
| 133.3                  | 14.8                                  | 11.6                             |
| 150.0                  | 16.6                                  | 13.0                             |

A general formula to compute the peak request rate with overlapping requests is:

$$\text{PEAKreq} = \text{freq} / [\text{entry} + (1 + \text{read\_ws}) + (1 + \text{write\_ws}) + \text{exit}]$$

where:

**Table 16-1071. Peak request formula operands**

| Operand | Description                                            |
|---------|--------------------------------------------------------|
| PEAKreq | Peak request rate                                      |
| freq    | System frequency                                       |
| entry   | Channel startup (4 cycles)                             |
| read_ws | Wait states seen during the system bus read data phase |

*Table continues on the next page...*

**Table 16-1071. Peak request formula operands (continued)**

| Operand  | Description                                             |
|----------|---------------------------------------------------------|
| write_ws | Wait states seen during the system bus write data phase |
| exit     | Channel shutdown (3 cycles)                             |

### 16.5.4.3 eDMA performance example

Consider a system with the following characteristics:

- Internal SRAM can be accessed with one wait-state when viewed from the system bus data phase
- All internal peripheral bus reads require two wait-states, and internal peripheral bus writes three wait-states viewed from the system bus data phase
- System operates at 150 MHz

For an SRAM to internal peripheral bus transfer,

$$\text{PEAKreq} = 150 \text{ MHz} / [ 4 + (1 + 1) + (1 + 3) + 3 ] \text{ cycles} = 11.5 \text{ Mreq/sec}$$

For an internal peripheral bus to SRAM transfer,

$$\text{PEAKreq} = 150 \text{ MHz} / [ 4 + (1 + 2) + (1 + 1) + 3 ] \text{ cycles} = 12.5 \text{ Mreq/sec}$$

Assuming an even distribution of the two transfer types, the average peak request rate would be:

$$\text{PEAKreq} = (11.5 \text{ Mreq/sec} + 12.5 \text{ Mreq/sec}) / 2 = 12.0 \text{ Mreq/sec}$$

The minimum number of cycles to perform a single read/write, zero wait states on the system bus, from a cold start where no channel is executing and eDMA is idle are:

- 11 cycles for a software, that is, a TCD<sub>n</sub>\_CSR[START] bit, request
- 12 cycles for a hardware, that is, an eDMA peripheral request signal, request

Two cycles account for the arbitration pipeline and one extra cycle on the hardware request resulting from the internal registering of the eDMA peripheral request signals. For the peak request rate calculations above, the arbitration and request registering is absorbed in or overlaps the previous executing channel.

**Note**

When channel linking or scatter/gather is enabled, a two cycle delay is imposed on the next channel selection and startup. This allows the link channel or the scatter/gather channel to be eligible and considered in the arbitration pool for next channel selection.

## 16.6 Initialization/application information

The following sections discuss initialization of the eDMA and programming considerations.

### 16.6.1 eDMA initialization

To initialize the eDMA:

1. Write to the CR if a configuration other than the default is desired.
2. Write the channel priority levels to the DCHPRI $_n$  registers if a configuration other than the default is desired.
3. Enable error interrupts in the EEI registers if so desired.
4. Write the 32-byte TCD for each channel that may request service.
5. Enable any hardware service requests via the ERQH and ERQL registers.
6. Request channel service via either:
  - Software: setting the TCD $_n$ \_CSR[START]
  - Hardware: slave device asserting its eDMA peripheral request signal

After any channel requests service, a channel is selected for execution based on the arbitration and priority levels written into the programmer's model. The eDMA engine reads the entire TCD, including the TCD control and status fields, as shown in the following table, for the selected channel into its internal address path module.

As the TCD is read, the first transfer is initiated on the internal bus, unless a configuration error is detected. Transfers from the source, as defined by TCD $_n$ \_SADDR, to the destination, as defined by TCD $_n$ \_DADDR, continue until the number of bytes specified by TCD $_n$ \_NBYTES are transferred.

When the transfer is complete, the eDMA engine's local TCD<sub>n</sub>\_SADDR, TCD<sub>n</sub>\_DADDR, and TCD<sub>n</sub>\_CITER are written back to the main TCD memory and any minor loop channel linking is performed, if enabled. If the major loop is exhausted, further post processing executes, such as interrupts, major loop channel linking, and scatter/gather operations, if enabled.

**Table 16-1072. TCD Control and Status fields**

| TCD <sub>n</sub> _CSR field name | Description                                                                                                             |
|----------------------------------|-------------------------------------------------------------------------------------------------------------------------|
| START                            | Control bit to start channel explicitly when using a software initiated DMA service (Automatically cleared by hardware) |
| ACTIVE                           | Status bit indicating the channel is currently in execution                                                             |
| DONE                             | Status bit indicating major loop completion (cleared by software when using a software initiated DMA service)           |
| D_REQ                            | Control bit to disable DMA request at end of major loop completion when using a hardware initiated DMA service          |
| BWC                              | Control bits for throttling bandwidth control of a channel                                                              |
| E_SG                             | Control bit to enable scatter-gather feature                                                                            |
| INT_HALF                         | Control bit to enable interrupt when major loop is half complete                                                        |
| INT_MAJ                          | Control bit to enable interrupt when major loop completes                                                               |

The following figure shows how each DMA request initiates one minor-loop transfer, or iteration, without CPU intervention. DMA arbitration can occur after each minor loop, and one level of minor loop DMA preemption is allowed. The number of minor loops in a major loop is specified by the beginning iteration count (BITER).

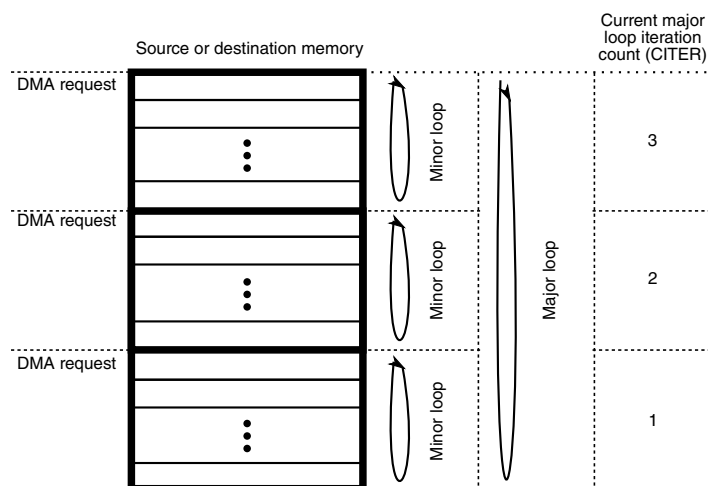


**Figure 16-1068. Example of multiple loop iterations**

The following figure lists the memory array terms and how the TCD settings interrelate.

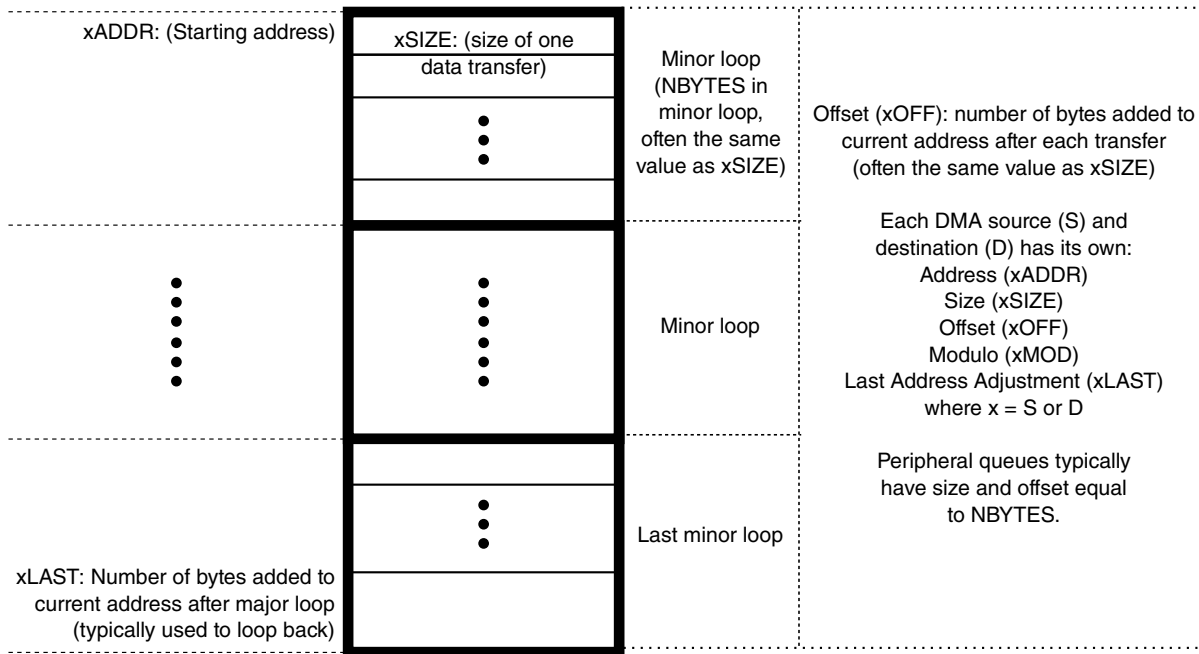


Figure 16-1069. Memory array terms

## 16.6.2 Programming errors

The eDMA performs various tests on the transfer control descriptor to verify consistency in the descriptor data. Most programming errors are reported on a per channel basis with the exception of two errors: Group Priority Error (ES[GPE]) and channel priority error (ES[CPE]).

For all error types other than group or channel priority errors, the channel number causing the error is recorded in the Error Status register (DMAx\_ES). If the error source is not removed before the next activation of the problem channel, the error is detected and recorded again.

Channel priority errors are identified within a group once that group has been selected as the active group. For example:

1. The eDMA is configured for fixed group and fixed channel arbitration modes.
2. Group 1 is the highest priority and all channels are unique in that group.
3. Group 0 is the next highest priority and has two channels with the same priority level.
4. If Group 1 has any service requests, those requests will be executed.
5. After all of Group 1 requests have completed, Group 0 will be the next active group.

6. If Group 0 has a service request, then an undefined channel in Group 0 will be selected and a channel priority error will occur.
7. This repeats until the all of Group 0 requests have been removed or a higher priority Group 1 request comes in.

In this sequence, for item 2, the eDMA acknowledge lines will assert only if the selected channel is requesting service via the eDMA peripheral request signal. If interrupts are enabled for all channels, the user will get an error interrupt, but the channel number for the ERR register and the error interrupt request line may be wrong because they reflect the selected channel. A group priority error is global and any request in any group will cause a group priority error.

If priority levels are not unique, when any channel requests service, a channel priority error is reported. The highest channel/group priority with an active request is selected, but the lowest numbered channel/group with that priority is selected by arbitration and executed by the eDMA engine. The hardware service request handshake signals, error interrupts, and error reporting is associated with the selected channel.

### **16.6.3 Arbitration mode considerations**

This section discusses arbitration considerations for the eDMA.

#### **16.6.3.1 Fixed group arbitration, Fixed channel arbitration**

In this mode, the channel service request from the highest priority channel in the highest priority group is selected to execute. If the eDMA is programmed so that the channels within one group use "fixed" priorities, and that group is assigned the highest "fixed" priority of all groups, that group can take all the bandwidth of the eDMA controller. That is, no other groups will be serviced if there is always at least one DMA request pending on a channel in the highest priority group when the controller arbitrates the next DMA request. The advantage of this scenario is that latency can be small for channels that need to be serviced quickly. Preemption is available in this scenario only.

#### **16.6.3.2 Round-robin group arbitration, fixed channel arbitration**

At the occurrence of one or more DMA requests from one or more groups, the channel with the highest priority from a specific group will be serviced first. Groups are serviced starting with the highest group number with an service request and rotating through to the lowest group number containing a service request.



Once the channel request is serviced, the group round robin algorithm will select the highest pending request from the next group in the round robin sequence. Servicing continues round robin, always servicing the highest priority channel in the next group in the sequence, or just skipping a group if it has no pending requests.

If a channel requests service at a rate that equals or exceeds the round robin service rate, then that channel will always be serviced before lower priority channels in the same group, and thus the lower priority channels will never be serviced. The advantage of this scenario is that no one group will consume all the eDMA bandwidth. The highest priority channel selection latency is potentially greater than fixed/fixed arbitration. Excessive request rates on high priority channels could prevent the servicing of lower priority channels in the same group.

### 16.6.3.3 Round-robin group arbitration, round-robin channel arbitration

Groups will be serviced as described in [Round-robin group arbitration, fixed channel arbitration](#), but this time channels will be serviced in channel number order. Only one channel is serviced from each requesting group for each round robin pass through the groups.

Within each group, channels are serviced starting with the highest channel number and rotating through to the lowest channel number without regard to channel priority levels.

Because channels are serviced in round robin manner, any channel that generates DMA requests faster than a combination of the group round robin service rate and the channel service rate for its group will not prevent the servicing of other channels in its group. Any DMA requests that are not serviced are simply lost, but at least one channel will be serviced.

This scenario ensures that all channels will be guaranteed service at some point, regardless of the request rates. However, the potential latency could be quite high. All channels are treated equally. Priority levels are not used in round robin/round robin mode.

### 16.6.3.4 Fixed group arbitration, Round-robin channel arbitration

The highest priority group with a request will be serviced. Lower priority groups will be serviced if no pending requests exist in the higher priority groups.

Within each group, channels are serviced starting with the highest channel number and rotating through to the lowest channel number without regard to the channel priority levels assigned within the group.

This scenario could cause the same bandwidth consumption problem as indicated in [Fixed group arbitration](#), [Fixed channel arbitration](#), but all the channels in the highest priority group will be serviced. Service latency will be short on the highest priority group, but could potentially be very much longer as the group priority decreases.

## 16.6.4 Performing DMA transfers

This section presents examples on how to perform DMA transfers with the eDMA.

### 16.6.4.1 Single request

To perform a simple transfer of  $n$  bytes of data with one activation, set the major loop to one ( $\text{TCDn\_CITER} = \text{TCDn\_BITER} = 1$ ). The data transfer begins after the channel service request is acknowledged and the channel is selected to execute. After the transfer is complete, the  $\text{TCDn\_CSR}[\text{DONE}]$  bit is set and an interrupt generates if properly enabled.

For example, the following TCD entry is configured to transfer 16 bytes of data. The eDMA is programmed for one iteration of the major loop transferring 16 bytes per iteration. The source memory has a byte wide memory port located at 0x1000. The destination memory has a 32-bit port located at 0x2000. The address offsets are programmed in increments to match the transfer size: one byte for the source and four bytes for the destination. The final source and destination addresses are adjusted to return to their beginning values.

```
TCDn_CITER = TCDn_BITER = 1
TCDn_NBYTES = 16
TCDn_SADDR = 0x1000
TCDn_SOFF = 1
TCDn_ATTR[SSIZE] = 0
TCDn_SLAST = -16
TCDn_DADDR = 0x2000
TCDn_DOFF = 4
TCDn_ATTR[DSIZE] = 2
TCDn_DLAST_SGA = -16
TCDn_CSR[INT_MAJ] = 1
TCDn_CSR[START] = 1 (Should be written last after all other fields have been initialized)
All other TCDn fields = 0
```

This generates the following event sequence:

1. User write to the  $\text{TCDn\_CSR}[\text{START}]$  bit requests channel service.

2. The channel is selected by arbitration for servicing.
3. eDMA engine writes:  $\text{TCDn\_CSR}[\text{DONE}] = 0$ ,  $\text{TCDn\_CSR}[\text{START}] = 0$ ,  $\text{TCDn\_CSR}[\text{ACTIVE}] = 1$ .
4. eDMA engine reads: channel TCD data from local memory to internal register file.
5. The source-to-destination transfers are executed as follows:
  - a. Read byte from location 0x1000, read byte from location 0x1001, read byte from 0x1002, read byte from 0x1003.
  - b. Write 32-bits to location 0x2000 → first iteration of the minor loop.
  - c. Read byte from location 0x1004, read byte from location 0x1005, read byte from 0x1006, read byte from 0x1007.
  - d. Write 32-bits to location 0x2004 → second iteration of the minor loop.
  - e. Read byte from location 0x1008, read byte from location 0x1009, read byte from 0x100A, read byte from 0x100B.
  - f. Write 32-bits to location 0x2008 → third iteration of the minor loop.
  - g. Read byte from location 0x100C, read byte from location 0x100D, read byte from 0x100E, read byte from 0x100F.
  - h. Write 32-bits to location 0x200C → last iteration of the minor loop → major loop complete.
6. The eDMA engine writes:  $\text{TCDn\_SADDR} = 0x1000$ ,  $\text{TCDn\_DADDR} = 0x2000$ ,  $\text{TCDn\_CITER} = 1$  ( $\text{TCDn\_BITER}$ ).
7. The eDMA engine writes:  $\text{TCDn\_CSR}[\text{ACTIVE}] = 0$ ,  $\text{TCDn\_CSR}[\text{DONE}] = 1$ ,  $\text{INT}[n] = 1$ .
8. The channel retires and the eDMA goes idle or services the next channel.

### 16.6.4.2 Multiple requests

The following example transfers 32 bytes via two hardware requests, but is otherwise the same as the previous example. The only fields that change are the major loop iteration count and the final address offsets. The eDMA is programmed for two iterations of the major loop transferring 16 bytes per iteration. After the channel's hardware requests are enabled in the ERQ register, the slave device initiates channel service requests.

```
TCDn_CITER = TCDn_BITER = 2  
TCDn_SLAST = -32  
TCDn_DLAST_SGA = -32
```

This would generate the following sequence of events:

1. First hardware, that is, eDMA peripheral, request for channel service.
2. The channel is selected by arbitration for servicing.
3. eDMA engine writes:  $\text{TCDn\_CSR}[\text{DONE}] = 0$ ,  $\text{TCDn\_CSR}[\text{START}] = 0$ ,  $\text{TCDn\_CSR}[\text{ACTIVE}] = 1$ .
4. eDMA engine reads: channel  $\text{TCDn}$  data from local memory to internal register file.
5. The source to destination transfers are executed as follows:
  - a. Read byte from location 0x1000, read byte from location 0x1001, read byte from 0x1002, read byte from 0x1003.
  - b. Write 32-bits to location 0x2000 → first iteration of the minor loop.
  - c. Read byte from location 0x1004, read byte from location 0x1005, read byte from 0x1006, read byte from 0x1007.
  - d. Write 32-bits to location 0x2004 → second iteration of the minor loop.
  - e. Read byte from location 0x1008, read byte from location 0x1009, read byte from 0x100A, read byte from 0x100B.
  - f. Write 32-bits to location 0x2008 → third iteration of the minor loop.
  - g. Read byte from location 0x100C, read byte from location 0x100D, read byte from 0x100E, read byte from 0x100F.
  - h. Write 32-bits to location 0x200C → last iteration of the minor loop.
6. eDMA engine writes:  $\text{TCDn\_SADDR} = 0x1010$ ,  $\text{TCDn\_DADDR} = 0x2010$ ,  $\text{TCDn\_CITER} = 1$ .
7. eDMA engine writes:  $\text{TCDn\_CSR}[\text{ACTIVE}] = 0$ .
8. The channel retires → one iteration of the major loop. The eDMA goes idle or services the next channel.
9. Second hardware, that is, eDMA peripheral, requests channel service.
10. The channel is selected by arbitration for servicing.
11. eDMA engine writes:  $\text{TCDn\_CSR}[\text{DONE}] = 0$ ,  $\text{TCDn\_CSR}[\text{START}] = 0$ ,  $\text{TCDn\_CSR}[\text{ACTIVE}] = 1$ .

12. eDMA engine reads: channel TCD data from local memory to internal register file.
13. The source to destination transfers are executed as follows:
  - a. Read byte from location 0x1010, read byte from location 0x1011, read byte from 0x1012, read byte from 0x1013.
  - b. Write 32-bits to location 0x2010 → first iteration of the minor loop.
  - c. Read byte from location 0x1014, read byte from location 0x1015, read byte from 0x1016, read byte from 0x1017.
  - d. Write 32-bits to location 0x2014 → second iteration of the minor loop.
  - e. Read byte from location 0x1018, read byte from location 0x1019, read byte from 0x101A, read byte from 0x101B.
  - f. Write 32-bits to location 0x2018 → third iteration of the minor loop.
  - g. Read byte from location 0x101C, read byte from location 0x101D, read byte from 0x101E, read byte from 0x101F.
  - h. Write 32-bits to location 0x201C → last iteration of the minor loop → major loop complete.
14. eDMA engine writes:  $\text{TCDn\_SADDR} = 0x1000$ ,  $\text{TCDn\_DADDR} = 0x2000$ ,  $\text{TCDn\_CITER} = 2$  ( $\text{TCDn\_BITER}$ ).
15. eDMA engine writes:  $\text{TCDn\_CSR}[\text{ACTIVE}] = 0$ ,  $\text{TCDn\_CSR}[\text{DONE}] = 1$ ,  $\text{INT}[n] = 1$ .
16. The channel retires → major loop complete. The eDMA goes idle or services the next channel.

### 16.6.4.3 Using the modulo feature

The modulo feature of the eDMA provides the ability to implement a circular data queue in which the size of the queue is a power of 2. MOD is a 5-bit field for the source and destination in the TCD, and it specifies which lower address bits increment from their original value after the address+offset calculation. All upper address bits remain the same as in the original value. A setting of 0 for this field disables the modulo feature.

The following table shows how the transfer addresses are specified based on the setting of the MOD field. Here a circular buffer is created where the address wraps to the original value while the 28 upper address bits (0x1234567x) retain their original value. In this example the source address is set to 0x12345670, the offset is set to 4 bytes and the MOD field is set to 4, allowing for a 2<sup>4</sup> byte (16-byte) size queue.

**Table 16-1073. Modulo example**

| Transfer Number | Address    |
|-----------------|------------|
| 1               | 0x12345670 |
| 2               | 0x12345674 |
| 3               | 0x12345678 |
| 4               | 0x1234567C |
| 5               | 0x12345670 |
| 6               | 0x12345674 |

## 16.6.5 Monitoring transfer descriptor status

This section discusses how to monitor eDMA status.

### 16.6.5.1 Testing for minor loop completion

There are two methods to test for minor loop completion when using software initiated service requests. The first is to read the TCD<sub>n</sub>\_CITER field and test for a change. Another method may be extracted from the sequence shown below. The second method is to test the TCD<sub>n</sub>\_CSR[START] bit and the TCD<sub>n</sub>\_CSR[ACTIVE] bit. The minor-loop-complete condition is indicated by both bits reading zero after the TCD<sub>n</sub>\_CSR[START] was set. Polling the TCD<sub>n</sub>\_CSR[ACTIVE] bit may be inconclusive, because the active status may be missed if the channel execution is short in duration.

The TCD status bits execute the following sequence for a software activated channel:

| Stage | TCD <sub>n</sub> _CSR bits |        |      | State                                            |
|-------|----------------------------|--------|------|--------------------------------------------------|
|       | START                      | ACTIVE | DONE |                                                  |
| 1     | 1                          | 0      | 0    | Channel service request via software             |
| 2     | 0                          | 1      | 0    | Channel is executing                             |
| 3a    | 0                          | 0      | 0    | Channel has completed the minor loop and is idle |
| 3b    | 0                          | 0      | 1    | Channel has completed the major loop and is idle |

The best method to test for minor-loop completion when using hardware, that is, peripheral, initiated service requests is to read the `TCDn_CITER` field and test for a change. The hardware request and acknowledge handshake signals are not visible in the programmer's model.

The TCD status bits execute the following sequence for a hardware-activated channel:

| Stage | TCDn_CSR bits |        |      | State                                                              |
|-------|---------------|--------|------|--------------------------------------------------------------------|
|       | START         | ACTIVE | DONE |                                                                    |
| 1     | 0             | 0      | 0    | Channel service request via hardware (peripheral request asserted) |
| 2     | 0             | 1      | 0    | Channel is executing                                               |
| 3a    | 0             | 0      | 0    | Channel has completed the minor loop and is idle                   |
| 3b    | 0             | 0      | 1    | Channel has completed the major loop and is idle                   |

For both activation types, the major-loop-complete status is explicitly indicated via the `TCDn_CSR[DONE]` bit.

The `TCDn_CSR[START]` bit is cleared automatically when the channel begins execution regardless of how the channel activates.

### 16.6.5.2 Reading the transfer descriptors of active channels

The eDMA reads back the true `TCDn_SADDR`, `TCDn_DADDR`, and `TCDn_NBYTES` values if read while a channel executes. The true values of the `SADDR`, `DADDR`, and `NBYTES` are the values the eDMA engine currently uses in its internal register file and not the values in the TCD local memory for that channel. The addresses, `SADDR` and `DADDR`, and `NBYTES`, which decrement to zero as the transfer progresses, can give an indication of the progress of the transfer. All other values are read back from the TCD local memory.

### 16.6.5.3 Checking channel preemption status

Preemption is available only when fixed arbitration is selected for both group and channel arbitration modes. A preemptive situation is one in which a preempt-enabled channel runs and a higher priority request becomes active. When the eDMA engine is not operating in fixed group, fixed channel arbitration mode, the determination of the actively running relative priority outstanding requests become undefined. Channel and/or group priorities are treated as equal, that is, constantly rotating, when Round-Robin Arbitration mode is selected.

The TCD $n$ \_CSR[ACTIVE] bit for the preempted channel remains asserted throughout the preemption. The preempted channel is temporarily suspended while the preempting channel executes one major loop iteration. If two TCD $n$ \_CSR[ACTIVE] bits are set simultaneously in the global TCD map, a higher priority channel is actively preempting a lower priority channel.

## 16.6.6 Channel Linking

Channel linking (or chaining) is a mechanism where one channel sets the TCD $n$ \_CSR[START] bit of another channel (or itself), therefore initiating a service request for that channel. When properly enabled, the EDMA engine automatically performs this operation at the major or minor loop completion.

The minor loop channel linking occurs at the completion of the minor loop (or one iteration of the major loop). The TCD $n$ \_CITER[E\_LINK] field determines whether a minor loop link is requested. When enabled, the channel link is made after each iteration of the major loop except for the last. When the major loop is exhausted, only the major loop channel link fields are used to determine if a channel link should be made. For example, the initial fields of:

```
TCDn_CITER[E_LINK] = 1
TCDn_CITER[LINKCH] = 0xC
TCDn_CITER[CITER] value = 0x4
TCDn_CSR[MAJOR_E_LINK] = 1
TCDn_CSR[MAJOR_LINKCH] = 0x7
```

executes as:

1. Minor loop done → set TCD12\_CSR[START] bit
2. Minor loop done → set TCD12\_CSR[START] bit
3. Minor loop done → set TCD12\_CSR[START] bit
4. Minor loop done, major loop done → set TCD7\_CSR[START] bit

When minor loop linking is enabled (TCD $n$ \_CITER[E\_LINK] = 1), the TCD $n$ \_CITER[CITER] field uses a nine bit vector to form the current iteration count. When minor loop linking is disabled (TCD $n$ \_CITER[E\_LINK] = 0), the TCD $n$ \_CITER[CITER] field uses a 15-bit vector to form the current iteration count. The bits associated with the TCD $n$ \_CITER[LINKCH] field are concatenated onto the CITER value to increase the range of the CITER.



## Note

The TCD $n$ \_CITER[E\_LINK] bit and the TCD $n$ \_BITER[E\_LINK] bit must equal or a configuration error is reported. The CITER and BITER vector widths must be equal to calculate the major loop, half-way done interrupt point.

The following table summarizes how a DMA channel can link to another DMA channel, i.e, use another channel's TCD, at the end of a loop.

**Table 16-1074. Channel Linking Parameters**

| Desired Link Behavior     | TCD Control Field Name | Description                                                                    |
|---------------------------|------------------------|--------------------------------------------------------------------------------|
| Link at end of Minor Loop | CITER[E_LINK]          | Enable channel-to-channel linking on minor loop completion (current iteration) |
|                           | CITER[LINKCH]          | Link channel number when linking at end of minor loop (current iteration)      |
| Link at end of Major Loop | CSR[MAJOR_E_LINK]      | Enable channel-to-channel linking on major loop completion                     |
|                           | CSR[MAJOR_LINKCH]      | Link channel number when linking at end of major loop                          |

## 16.6.7 Dynamic programming

This section provides recommended methods to change the programming model during channel execution.

### 16.6.7.1 Dynamically changing the channel priority

The following two options are recommended for dynamically changing channel priority levels:

1. Switch to Round-Robin Channel Arbitration mode, change the channel priorities, then switch back to Fixed Arbitration mode,
2. Disable all the channels within a group, change the channel priorities within that group only, then enable the appropriate channels.

The following two options are available for dynamically changing group priority levels:

1. Switch to Round-Robin Group Arbitration mode, change the group priorities, then switch back to Fixed Arbitration mode,
2. Disable all channels, change the group priorities, then enable the appropriate channels.

### 16.6.7.2 Dynamic channel linking

Dynamic channel linking is the process of setting the TCD.major.e\_link bit during channel execution (see the diagram in [TCD structure](#)). This bit is read from the TCD local memory at the end of channel execution, thus allowing the user to enable the feature during channel execution.

Because the user is allowed to change the configuration during execution, a coherency model is needed. Consider the scenario where the user attempts to execute a dynamic channel link by enabling the TCD.major.e\_link bit at the same time the eDMA engine is retiring the channel. The TCD.major.e\_link would be set in the programmer's model, but it would be unclear whether the actual link was made before the channel retired.

The following coherency model is recommended when executing a dynamic channel link request.

1. Write 1 to the TCD.major.e\_link bit.
2. Read back the TCD.major.e\_link bit.
3. Test the TCD.major.e\_link request status:
  - If TCD.major.e\_link = 1, the dynamic link attempt was successful.
  - If TCD.major.e\_link = 0, the attempted dynamic link did not succeed (the channel was already retiring).

For this request, the TCD local memory controller forces the TCD.major.e\_link bit to zero on any writes to a channel's TCD.word7 after that channel's TCD.done bit is set, indicating the major loop is complete.

#### NOTE

The user must clear the TCD.done bit before writing the TCD.major.e\_link bit. The TCD.done bit is cleared automatically by the eDMA engine after a channel begins execution.

### 16.6.7.3 Dynamic scatter/gather

Scatter/gather is the process of automatically loading a new TCD into a channel. It allows a DMA channel to use multiple TCDs; this enables a DMA channel to scatter the DMA data to multiple destinations or gather it from multiple sources. When scatter/gather is enabled and the channel has finished its major loop, a new TCD is fetched from system memory and loaded into that channel's descriptor location in eDMA programmer's model, thus replacing the current descriptor.

Because the user is allowed to change the configuration during execution, a coherency model is needed. Consider the scenario where the user attempts to execute a dynamic scatter/gather operation by enabling the TCD.e\_sg bit at the same time the eDMA engine is retiring the channel. The TCD.e\_sg would be set in the programmer's model, but it would be unclear whether the actual scatter/gather request was honored before the channel retired.

Two methods for this coherency model are shown in the following subsections. Method 1 has the advantage of reading the major.linkch field and the e\_sg bit with a single read. For both dynamic channel linking and scatter/gather requests, the TCD local memory controller forces the TCD.major.e\_link and TCD.e\_sg bits to zero on any writes to a channel's TCD.word7 if that channel's TCD.done bit is set indicating the major loop is complete.

### NOTE

The user must clear the TCD.done bit before writing the TCD.major.e\_link or TCD.e\_sg bits. The TCD.done bit is cleared automatically by the eDMA engine after a channel begins execution.

#### 16.6.7.3.1 Method 1 (channel not using major loop channel linking)

For a channel not using major loop channel linking, the coherency model described here may be used for a dynamic scatter/gather request.

When the TCD.major.e\_link bit is zero, the TCD.major.linkch field is not used by the eDMA. In this case, the TCD.major.linkch bits may be used for other purposes. This method uses the TCD.major.linkch field as a TCD identification (ID).

1. When the descriptors are built, write a unique TCD ID in the TCD.major.linkch field for each TCD associated with a channel using dynamic scatter/gather.
2. Write 1b to the TCD.d\_req bit.

Should a dynamic scatter/gather attempt fail, setting the TCD.d\_req bit will prevent a future hardware activation of this channel. This stops the channel from executing with a destination address (daddr) that was calculated using a scatter/gather address (written in the next step) instead of a dlast final offset value.

3. Write the TCD.dlast\_sga field with the scatter/gather address.
4. Write 1b to the TCD.e\_sg bit.
5. Read back the 16 bit TCD control/status field.

6. Test the TCD.e\_sg request status and TCD.major.linkch value:

If e\_sg = 1b, the dynamic link attempt was successful.

If e\_sg = 0b and the major.linkch (ID) did not change, the attempted dynamic link did not succeed (the channel was already retiring).

If e\_sg = 0b and the major.linkch (ID) changed, the dynamic link attempt was successful (the new TCD's e\_sg value cleared the e\_sg bit).

### 16.6.7.3.2 Method 2 (channel using major loop channel linking)

For a channel using major loop channel linking, the coherency model described here may be used for a dynamic scatter/gather request. This method uses the TCD.dlast\_sga field as a TCD identification (ID).

1. Write 1b to the TCD.d\_req bit.

Should a dynamic scatter/gather attempt fail, setting the d\_req bit will prevent a future hardware activation of this channel. This stops the channel from executing with a destination address (daddr) that was calculated using a scatter/gather address (written in the next step) instead of a dlast final offset value.

2. Write the TCD.dlast\_sga field with the scatter/gather address.

3. Write 1b to the TCD.e\_sg bit.

4. Read back the TCD.e\_sg bit.

5. Test the TCD.e\_sg request status:

If e\_sg = 1b, the dynamic link attempt was successful.

If e\_sg = 0b, read the 32 bit TCD dlast\_sga field.

If e\_sg = 0b and the dlast\_sga did not change, the attempted dynamic link did not succeed (the channel was already retiring).

If e\_sg = 0b and the dlast\_sga changed, the dynamic link attempt was successful (the new TCD's e\_sg value cleared the e\_sg bit).

### 16.6.8 Suspend/resume a DMA channel with active hardware service requests

The DMA allows the user to move data from memory or peripheral registers to another location in memory or peripheral registers without CPU interaction. Once the DMA and peripherals have been configured and are active, it is rare to suspend a peripheral's service request dynamically. In this scenario, there are certain restrictions to disabling a DMA hardware service request. For coherency, a specific procedure must be followed. This section provides guidance on how to coherently suspend and resume a Direct Memory Access (DMA) channel when the DMA is triggered by a slave module such as the Serial Peripheral Interface (DSPI), ADC, or other module.

#### 16.6.8.1 Suspend an active DMA channel

To suspend an active DMA channel:

1. Stop the DMA service request at the peripheral first. Confirm it has been disabled by reading back the appropriate register in the peripheral.
2. Check to ensure there is no service request to the DMA channel being suspended. Then disable the hardware service request by clearing the ERQ bit on the appropriate DMA channel.

#### 16.6.8.2 Resume a DMA channel

To resume a DMA channel:

1. Enable the DMA service request on the appropriate channel by setting its ERQ bit.
2. Enable the DMA service request at the peripheral.

For example, assume a DSPI module is set as a master for transmitting data via a DMA service request when the DSPI\_TXFIFO has an empty slot. The DMA will transfer the next command and data to the TXFIFO upon the request. If the user needs to suspend the DMA/DSPI transfer loop, perform the following steps:

1. Disable the DMA service request at the source by writing 0 to DSPI\_RSER[TFFF\_RE] . Confirm that DSPI\_RSER[TFFF\_RE] is 0.
2. Ensure there is no DMA service request from the DSPI by verifying that DMA\_HRS[HRS<sub>n</sub>] is 0 for the appropriate channel. If no service request is present, disable the DMA channel by clearing the channel's ERQ bit. If a service request is present, wait until the request has been processed and the HRS bit reads zero.



# Chapter 17

## Error Injection Module (EIM)

### 17.1 Chip-specific EIM information

This chip has one instance of the EIM.

#### 17.1.1 EIM channels

The EIM on this chip supports 6 channels. The number of each channel corresponds to:

- A field in the EIM's EICHEN register
- An EICHD $n$ \_Word0 register
- An EICHD $n$ \_Word1 register
- An EICHD $n$ \_Word2 register

#### 17.1.2 EIM channel assignments

Each EIM channel corresponds to a specific RAM array target. The EIM channel can inject errors on accesses to that specific RAM array. The following table shows the channel assignments.

**Table 17-1. RAM array targets of EIM channels**

| EIM channel | RAM array target |
|-------------|------------------|
| 0           | PRAMC_0          |
| 1           | PRAMC_1          |
| 2           | FEC MIB          |
| 3           | FEC RIF          |
| 4           | eDMA_A TCD       |
| 5           | eDMA_B TCD       |

### 17.1.3 EIM\_EICHDN\_Word register bit mapping

The following table shows the implemented bits of the [Error Injection Channel Descriptor, Word0 \(EIM\\_EICHDN\\_WORD0\)](#), [Error Injection Channel Descriptor, Word1 \(EIM\\_EICHDN\\_WORD1\)](#), and [Error Injection Channel Descriptor, Word2 \(EIM\\_EICHDN\\_WORD2\)](#) registers for each channel.

| Off set | 0                                     | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9        | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28                                 | 29 | 30 | 31 |
|---------|---------------------------------------|---|---|---|---|---|---|---|---|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------------------------------|----|----|----|
| 100     | PRAMC_0 CHKBIT_MASK                   |   |   |   |   |   |   |   |   | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                                    |    |    |    |
| 104     | PRAMC_0 B0_3DATA_MASK (bits 63-32)    |   |   |   |   |   |   |   |   |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                                    |    |    |    |
| 108     | PRAMC_0 B4_7DATA_MASK (bits 31-0)     |   |   |   |   |   |   |   |   |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                                    |    |    |    |
| 200     | PRAMC_1 CHKBIT_MASK                   |   |   |   |   |   |   |   |   | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                                    |    |    |    |
| 204     | PRAMC_1 B0_3DATA_MASK (bits 63-32)    |   |   |   |   |   |   |   |   |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                                    |    |    |    |
| 208     | PRAMC_1 B4_7DATA_MASK (bits 31-0)     |   |   |   |   |   |   |   |   |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                                    |    |    |    |
| 300     | FEC MIB CHKBIT_MASK                   |   |   |   |   |   |   |   |   | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                                    |    |    |    |
| 304     | FEC MIB B0_3DATA_MASK (bits 31-0)     |   |   |   |   |   |   |   |   |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                                    |    |    |    |
| 308     | Reserved                              |   |   |   |   |   |   |   |   |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                                    |    |    |    |
| 400     | FEC RIF CHKBIT_MASK                   |   |   |   |   |   |   |   |   | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                                    |    |    |    |
| 404     | Reserved                              |   |   |   |   |   |   |   |   |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | FEC RIF B0_3DATA_MASK (bits 35-32) |    |    |    |
| 408     | FEC RIF B4_7DATA_MASK (bits 31-0)     |   |   |   |   |   |   |   |   |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                                    |    |    |    |
| 500     | eDMA_A TCD CHKBIT_MASK                |   |   |   |   |   |   |   |   | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                                    |    |    |    |
| 504     | eDMA_A TCD B0_3DATA_MASK (bits 63-32) |   |   |   |   |   |   |   |   |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                                    |    |    |    |
| 508     | eDMA_A TCD B4_7DATA_MASK (bits 31-0)  |   |   |   |   |   |   |   |   |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                                    |    |    |    |
| 600     | eDMA_B TCD CHKBIT_MASK                |   |   |   |   |   |   |   |   | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                                    |    |    |    |
| 604     | eDMA_B TCD B0_3DATA_MASK (bits 63-32) |   |   |   |   |   |   |   |   |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                                    |    |    |    |
| 608     | eDMA_B TCD B4_7DATA_MASK (bits 31-0)  |   |   |   |   |   |   |   |   |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                                    |    |    |    |

## 17.2 Introduction

The Error Injection module is mainly used for diagnostic purposes. It provides a method for diagnostic coverage of the peripheral memories. See the chip-specific EIM information to determine which peripheral memories are supported by this method.



## 17.2.1 Overview

The Error Injection Module (EIM) provides support for inducing single-bit and multi-bit inversions on read data when accessing peripheral RAMs. Injecting faults on memory accesses can be used to exercise the SEC-DED ECC function of the related system.

### NOTE

The following diagram shows an example EIM implementation with a 64-bit read data bus and an 8-bit checkbit bus.

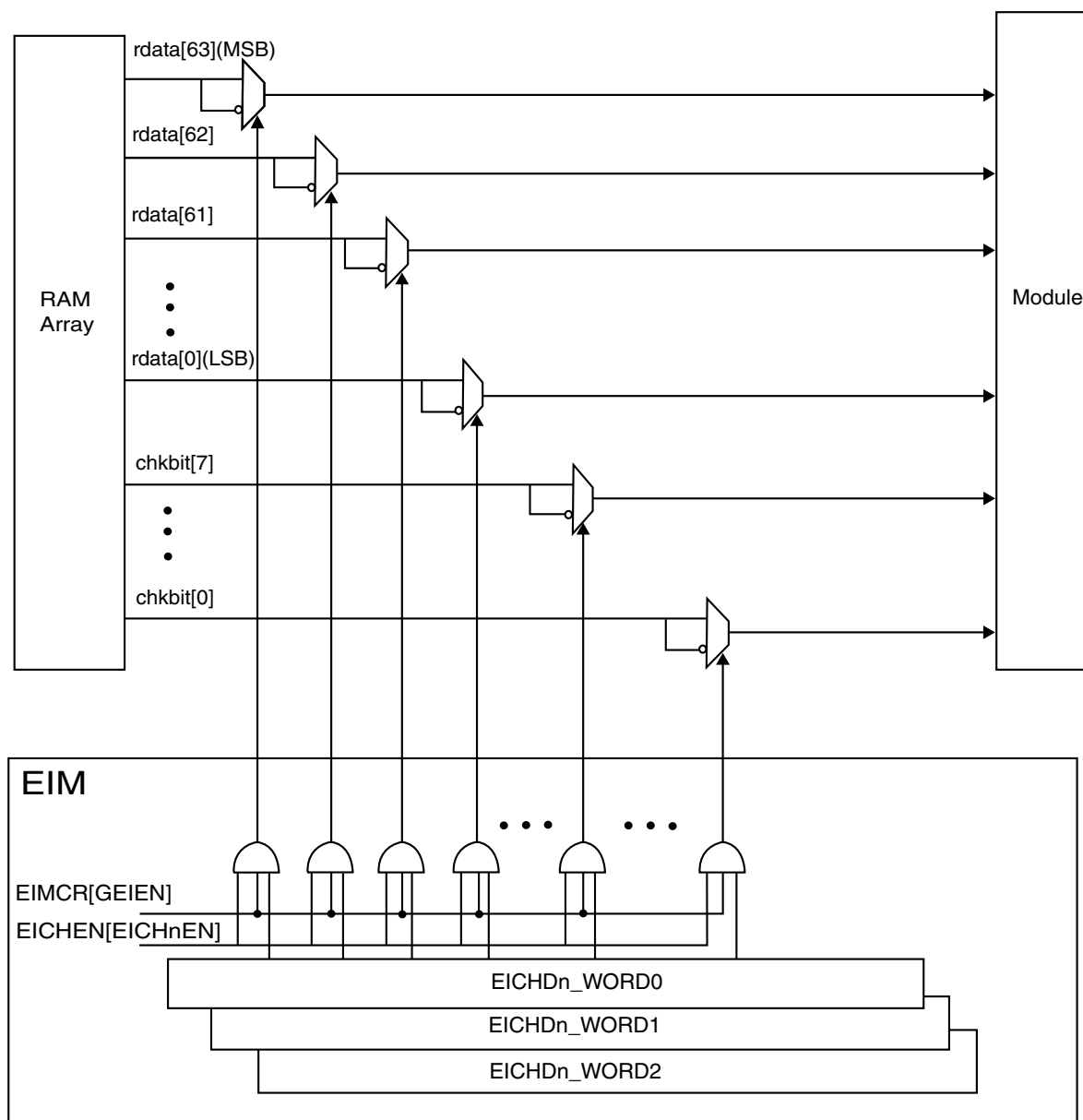


Figure 17-1. EIM functional block diagram (64-bit read data bus and 8-bit checkbit bus)

## 17.2.2 Features

The EIM includes these features:

- Supports 6 error injection channels
- Protection against accidental enable and reconfiguration error injection function via two-stage enable mechanism

## 17.3 Memory map and register definition

The EIM provides an IPS programming model mapped to an on-platform peripheral slot.

### Programming model access

All system bus masters can access the programming model:

- Only in supervisor mode
- Using only 32-bit (word) accesses

Any of the following attempted references to the programming model generates an IPS error termination:

- In user mode
- Using non-32-bit access sizes
- To undefined (reserved) addresses

Attempted updates to the programming model while the EIM is in the midst of an operation will result in non-deterministic behavior.

### Error injection channel descriptor: function and structure

Each error injection channel descriptor:

- Specifies a mask that defines which bits of the read data and checkbit bus from target RAM are inverted on a read access.
- Consists of a 128-bit (16-byte) structure, composed of four 32-bit words, in the EIM programming model.

- Each descriptor consists of Error Injection Channel Descriptor Word0-1 (EICHD $n$ \_WORD0-1) and, depending on the implementation, Word2 (EICHD $n$ \_WORD2).
- When Word2 is unused, the corresponding word in the 16-byte structure is unused.
- The fourth word in the 16-byte structure is always unused.

The multiple channel descriptors are organized sequentially.

### EIM memory map

| Address offset (hex) | Register name                                                | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|--------------------------------------------------------------|-----------------|--------|-------------|----------------------------|
| 0                    | Error Injection Module Configuration Register (EIM_EIMCR)    | 32              | R/W    | 0000_0000h  | <a href="#">17.3.1/720</a> |
| 4                    | Error Injection Channel Enable register (EIM_EICHEN)         | 32              | R/W    | 0000_0000h  | <a href="#">17.3.2/721</a> |
| 100                  | Error Injection Channel Descriptor, Word0 (EIM_EICHD0_WORD0) | 32              | R/W    | 0000_0000h  | <a href="#">17.3.3/723</a> |
| 104                  | Error Injection Channel Descriptor, Word1 (EIM_EICHD0_WORD1) | 32              | R/W    | 0000_0000h  | <a href="#">17.3.4/724</a> |
| 108                  | Error Injection Channel Descriptor, Word2 (EIM_EICHD0_WORD2) | 32              | R/W    | 0000_0000h  | <a href="#">17.3.5/724</a> |
| 200                  | Error Injection Channel Descriptor, Word0 (EIM_EICHD1_WORD0) | 32              | R/W    | 0000_0000h  | <a href="#">17.3.3/723</a> |
| 204                  | Error Injection Channel Descriptor, Word1 (EIM_EICHD1_WORD1) | 32              | R/W    | 0000_0000h  | <a href="#">17.3.4/724</a> |
| 208                  | Error Injection Channel Descriptor, Word2 (EIM_EICHD1_WORD2) | 32              | R/W    | 0000_0000h  | <a href="#">17.3.5/724</a> |
| 300                  | Error Injection Channel Descriptor, Word0 (EIM_EICHD2_WORD0) | 32              | R/W    | 0000_0000h  | <a href="#">17.3.3/723</a> |
| 304                  | Error Injection Channel Descriptor, Word1 (EIM_EICHD2_WORD1) | 32              | R/W    | 0000_0000h  | <a href="#">17.3.4/724</a> |
| 308                  | Error Injection Channel Descriptor, Word2 (EIM_EICHD2_WORD2) | 32              | R/W    | 0000_0000h  | <a href="#">17.3.5/724</a> |
| 400                  | Error Injection Channel Descriptor, Word0 (EIM_EICHD3_WORD0) | 32              | R/W    | 0000_0000h  | <a href="#">17.3.3/723</a> |
| 404                  | Error Injection Channel Descriptor, Word1 (EIM_EICHD3_WORD1) | 32              | R/W    | 0000_0000h  | <a href="#">17.3.4/724</a> |
| 408                  | Error Injection Channel Descriptor, Word2 (EIM_EICHD3_WORD2) | 32              | R/W    | 0000_0000h  | <a href="#">17.3.5/724</a> |
| 500                  | Error Injection Channel Descriptor, Word0 (EIM_EICHD4_WORD0) | 32              | R/W    | 0000_0000h  | <a href="#">17.3.3/723</a> |
| 504                  | Error Injection Channel Descriptor, Word1 (EIM_EICHD4_WORD1) | 32              | R/W    | 0000_0000h  | <a href="#">17.3.4/724</a> |
| 508                  | Error Injection Channel Descriptor, Word2 (EIM_EICHD4_WORD2) | 32              | R/W    | 0000_0000h  | <a href="#">17.3.5/724</a> |

Table continues on the next page...

## EIM memory map (continued)

| Address offset (hex) | Register name                                                | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|--------------------------------------------------------------|-----------------|--------|-------------|----------------------------|
| 600                  | Error Injection Channel Descriptor, Word0 (EIM_EICHD5_WORD0) | 32              | R/W    | 0000_0000h  | <a href="#">17.3.3/723</a> |
| 604                  | Error Injection Channel Descriptor, Word1 (EIM_EICHD5_WORD1) | 32              | R/W    | 0000_0000h  | <a href="#">17.3.4/724</a> |
| 608                  | Error Injection Channel Descriptor, Word2 (EIM_EICHD5_WORD2) | 32              | R/W    | 0000_0000h  | <a href="#">17.3.5/724</a> |

### 17.3.1 Error Injection Module Configuration Register (EIM\_EIMCR)

The EIM Configuration Register is used to globally enable/disable the error injection function.

Address: 0h base + 0h offset = 0h

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |       |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--|--|--|--|--|--|--|--|--|--|--|--|--|--|-------|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |       |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |       |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |       |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |       |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |       |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  | GEIEN |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |       |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |       |

#### EIM\_EIMCR field descriptions

| Field            | Description                                                                                                                                                                                     |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–30<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                         |
| 31<br>GEIEN      | Global Error Injection Enable<br><br>This bit globally enables or disables the error injection function of the EIM. This field is initialized by hardware reset.<br><br>0 Disabled<br>1 Enabled |

## 17.3.2 Error Injection Channel Enable register (EIM\_EICHEN)

Each field of the Error Injection Channel Enable register (EICHEN) is used to enable or disable the corresponding error injection channel.

### NOTE

To enable an error injection channel, the Global Error Injection Enable (EIMCR[GEIEN]) field must also be asserted.

Address: 0h base + 4h offset = 4h

|       |         |         |         |         |         |         |    |    |    |    |    |    |    |    |    |    |
|-------|---------|---------|---------|---------|---------|---------|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0       | 1       | 2       | 3       | 4       | 5       | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | EICH0EN | EICH1EN | EICH2EN | EICH3EN | EICH4EN | EICH5EN | 0  |    |    |    |    |    |    |    |    |    |
| W     |         |         |         |         |         |         |    |    |    |    |    |    |    |    |    |    |
| Reset | 0       | 0       | 0       | 0       | 0       | 0       | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| Bit   | 16      | 17      | 18      | 19      | 20      | 21      | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0       |         |         |         |         |         |    |    |    |    |    |    |    |    |    |    |
| W     |         |         |         |         |         |         |    |    |    |    |    |    |    |    |    |    |
| Reset | 0       | 0       | 0       | 0       | 0       | 0       | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

### EIM\_EICHEN field descriptions

| Field        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>EICH0EN | <p>Error Injection Channel 0 Enable</p> <p>This field enables the corresponding error injection channel. The Global Error Injection Enable (EIMCR[GEIEN]) field must also be asserted to enable error injection.</p> <p>After error injection is enabled, all subsequent read accesses incur one or more bit inversions as defined in the corresponding EICHn_WORD registers. Error injection remains in effect until the error injection channel is manually disabled via software.</p> <p>Any write to the corresponding EICHn_WORD registers clears the corresponding EICHEN[EICHnEN] field, disabling the error injection channel.</p> <p>0 Error injection is disabled on Error Injection Channel 0<br/>1 Error injection is enabled on Error Injection Channel 0</p> |
| 1<br>EICH1EN | <p>Error Injection Channel 1 Enable</p> <p>This field enables the corresponding error injection channel. The Global Error Injection Enable (EIMCR[GEIEN]) field must also be asserted to enable error injection.</p> <p>After error injection is enabled, all subsequent read accesses incur one or more bit inversions as defined in the corresponding EICHn_WORD registers. Error injection remains in effect until the error injection channel is manually disabled via software.</p> <p>Any write to the corresponding EICHn_WORD registers clears the corresponding EICHEN[EICHnEN] field, disabling the error injection channel.</p>                                                                                                                                 |

Table continues on the next page...

**EIM\_EICHEN field descriptions (continued)**

| Field        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|              | 0 Error injection is disabled on Error Injection Channel 1<br>1 Error injection is enabled on Error Injection Channel 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 2<br>EICH2EN | Error Injection Channel 2 Enable<br><br>This field enables the corresponding error injection channel. The Global Error Injection Enable (EIMCR[GEIEN]) field must also be asserted to enable error injection.<br><br>After error injection is enabled, all subsequent read accesses incur one or more bit inversions as defined in the corresponding EICHD <sub>n</sub> _WORD registers. Error injection remains in effect until the error injection channel is manually disabled via software.<br><br>Any write to the corresponding EICHD <sub>n</sub> _WORD registers clears the corresponding EICHEN[EICH <sub>n</sub> EN] field, disabling the error injection channel.<br><br>0 Error injection is disabled on Error Injection Channel 2<br>1 Error injection is enabled on Error Injection Channel 2 |
| 3<br>EICH3EN | Error Injection Channel 3 Enable<br><br>This field enables the corresponding error injection channel. The Global Error Injection Enable (EIMCR[GEIEN]) field must also be asserted to enable error injection.<br><br>After error injection is enabled, all subsequent read accesses incur one or more bit inversions as defined in the corresponding EICHD <sub>n</sub> _WORD registers. Error injection remains in effect until the error injection channel is manually disabled via software.<br><br>Any write to the corresponding EICHD <sub>n</sub> _WORD registers clears the corresponding EICHEN[EICH <sub>n</sub> EN] field, disabling the error injection channel.<br><br>0 Error injection is disabled on Error Injection Channel 3<br>1 Error injection is enabled on Error Injection Channel 3 |
| 4<br>EICH4EN | Error Injection Channel 4 Enable<br><br>This field enables the corresponding error injection channel. The Global Error Injection Enable (EIMCR[GEIEN]) field must also be asserted to enable error injection.<br><br>After error injection is enabled, all subsequent read accesses incur one or more bit inversions as defined in the corresponding EICHD <sub>n</sub> _WORD registers. Error injection remains in effect until the error injection channel is manually disabled via software.<br><br>Any write to the corresponding EICHD <sub>n</sub> _WORD registers clears the corresponding EICHEN[EICH <sub>n</sub> EN] field, disabling the error injection channel.<br><br>0 Error injection is disabled on Error Injection Channel 4<br>1 Error injection is enabled on Error Injection Channel 4 |
| 5<br>EICH5EN | Error Injection Channel 5 Enable<br><br>This field enables the corresponding error injection channel. The Global Error Injection Enable (EIMCR[GEIEN]) field must also be asserted to enable error injection.<br><br>After error injection is enabled, all subsequent read accesses incur one or more bit inversions as defined in the corresponding EICHD <sub>n</sub> _WORD registers. Error injection remains in effect until the error injection channel is manually disabled via software.<br><br>Any write to the corresponding EICHD <sub>n</sub> _WORD registers clears the corresponding EICHEN[EICH <sub>n</sub> EN] field, disabling the error injection channel.<br><br>0 Error injection is disabled on Error Injection Channel 5<br>1 Error injection is enabled on Error Injection Channel 5 |

*Table continues on the next page...*

**EIM\_EICHEN field descriptions (continued)**

| Field            | Description                                                                             |
|------------------|-----------------------------------------------------------------------------------------|
| 6–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |

**17.3.3 Error Injection Channel Descriptor, Word0  
(EIM\_EICHD<sub>n</sub>\_WORD0)**

The first word of the Error Injection Channel Descriptor defines a left-justified mask field (CHKBIT\_MASK) whose width is up to 8 bits. Each bit of CHKBIT\_MASK specifies whether the corresponding bit of the checkbit bus from the target RAM should be inverted or remain unmodified on read accesses. Successful writes to this field clear the corresponding error injection channel valid bit, EICHEN[EICH<sub>n</sub>EN].

Address: 0h base + 100h offset + (256d × i), where i=0d to 5d

|       |             |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0           | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | CHKBIT_MASK |   |   |   |   |   |   |   | 0 |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |             |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**EIM\_EICHD<sub>n</sub>\_WORD0 field descriptions**

| Field              | Description                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–7<br>CHKBIT_MASK | <p>Checkbit Mask</p> <p>This field defines a bit-mapped mask that specifies whether the corresponding bit of the checkbit bus from the target RAM should be inverted or remain unmodified.</p> <p>The width of the field for the channel is defined in the chip-specific EIM information.</p> <p>0 The corresponding bit of the checkbit bus remains unmodified.<br/>1 The corresponding bit of the checkbit bus is inverted.</p> |
| 8–31<br>Reserved   | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                |

### 17.3.4 Error Injection Channel Descriptor, Word1 (EIM\_EICH $D_n$ \_WORD1)

The second word of the Error Injection Channel Descriptor defines the bits in the mask corresponding to bytes 0–3 of the read data bus. Each bit specifies whether the corresponding bit of the read data bus from the target RAM should be inverted or remain unmodified on read accesses. Successful writes to this field clear the corresponding error injection channel valid field, EICHEN[EICH $n$ EN].

Address: 0h base + 104h offset + (256d × i), where i=0d to 5d

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**EIM\_EICH $D_n$ \_WORD1 field descriptions**

| Field                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>B0_3DATA_MASK | <p>Data Mask Bytes 0-3</p> <p>This field defines a bit-mapped mask that specifies whether the corresponding bit of the read data bus from the target RAM should be inverted or remain unmodified.</p> <p>For details about which bits in this field are implemented for the applicable channel and how the bits map to bytes 0-3 of the read data bus, see the chip-specific EIM information.</p> <p>0 The corresponding bit of bytes 0-3 on the read data bus remains unmodified</p> <p>1 The corresponding bit of bytes 0-3 on the read data bus is inverted.</p> |

### 17.3.5 Error Injection Channel Descriptor, Word2 (EIM\_EICH $D_n$ \_WORD2)

The third word of the Error Injection Channel Descriptor, when present, defines the bits in the mask corresponding to bytes 4–7 of the read data bus. Each bit specifies whether the corresponding bit of the read data bus from the target RAM should be inverted or remain unmodified on read accesses. Successful writes to this field clear the corresponding error injection channel valid field, EICHEN[EICH $n$ EN].

Address: 0h base + 108h offset + (256d × i), where i=0d to 5d

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |



**EIM\_EICHD<sub>n</sub>\_WORD2 field descriptions**

| Field                     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>B4_7DATA_<br>MASK | <p>Data Mask Bytes 4-7</p> <p>This field defines a bit-mapped mask that specifies whether the corresponding bit of the read data bus from the target RAM should be inverted or remain unmodified.</p> <p>For details about which bits in this field are implemented for the applicable channel and how the bits map to bytes 4-7 of the read data bus, see the chip-specific EIM information.</p> <p>0 The corresponding bit of bytes 4-7 on the read data bus remains unmodified.</p> <p>1 The corresponding bit of bytes 4-7 on the read data bus is inverted.</p> |

## 17.4 Functional description

The EIM provides protection against accidental enabling and reconfiguration of the error injection function by enforcing a two-stage enablement mechanism. To properly enable the error injection mechanism for a channel:

- Write 1 to the EICHEN[EICH<sub>n</sub>EN] field, where *n* denotes the channel number.
- Write 1 to EIMCR[GEIEN].

### NOTE

When the use case for a channel requires writing any EICHD<sub>n</sub>\_WORD register, write the EICHD<sub>n</sub>\_WORD register before executing the two-stage enablement mechanism. A successful write to any EICHD<sub>n</sub>\_WORD register clears the corresponding EICHEN[EICH<sub>n</sub>EN] field.

The EIM supports 6 error injection channels. Each channel:

- is assigned to a single memory array interface.
- intercepts the assigned memory read data bus and checkbit bus and injects errors by inverting the value transmitted for selected bits on each bus line.

On a memory read access, the applicable EICHD<sub>n</sub>\_WORD registers define which bits of the read data and/or checkbit bus to invert.

Figure 17-1 depicts the interception and override of a 64-bit read data bus and an 8-bit checkbit data bus for an example memory array.

### 17.4.1 Error injection scenarios

The EIM supports these cases of error injection:

- To generate a single-bit error, invert only 1 bit of the CHKBIT\_MASK or DATA\_MASK in the EICHD $n$ \_WORD registers.
- To generate a multi-bit error, invert only 2 bits of the CHKBIT\_MASK or DATA\_MASK in the EICHD $n$ \_WORD registers.

#### NOTE

An attempt to invert more than 2 bits in one operation might result in undefined behavior.

# Chapter 18

## Error Reporting Module (ERM)

### 18.1 Chip-specific ERM information

This chip has one instance of the ERM.

#### 18.1.1 ERM memory channels

The ERM on this chip supports 20 memory channels. The number of each memory channel corresponds to:

- A field in the ERM's CR0, CR1, or CR2 register
- A field in the ERM's SR0, SR1, or SR2 register
- An ERM\_EAR<sub>n</sub> register
- An ERM\_SYN<sub>n</sub> register

#### 18.1.2 Sources of memory error events

Each ERM channel corresponds to a source of potential memory error events. The following table shows the channel assignments.

**Table 18-1. Memory error event sources for ERM channels**

| ERM channel | Memory event source     |
|-------------|-------------------------|
| 0           | PRAMC_0                 |
| 1           | PRAMC_1                 |
| 2           | eDMA_A TCD <sup>1</sup> |
| 3           | eDMA_B TCD <sup>1</sup> |
| 4           | FEC MIB <sup>1</sup>    |

*Table continues on the next page...*

**Table 18-1. Memory error event sources for ERM channels (continued)**

| ERM channel | Memory event source  |
|-------------|----------------------|
| 5           | FEC RIF <sup>1</sup> |
| 6           | PFLASH port 0        |
| 7           | PFLASH port 1        |
| 8           | AIPS_0               |
| 9           | AIPS_1               |
| 10          | FEC e2eECC           |
| 11          | CSE                  |
| 13          | Core0 instruction    |
| 14          | Core0 data           |
| 15          | Core1 instruction    |
| 16          | Core1 data           |
| 17          | eDMA_A e2eECC        |
| 18          | eDMA_B e2eECC        |

1. For these memory elements, the error address captured is the relative array address and not the system address.

### 18.1.3 Managing single-bit faults of TCD RAM for eDMA\_A

When a single-bit ECC event occurs during a read of the TCD RAM for eDMA\_A, the fault is correctly identified and forwarded to the ERM. However, the fault is not corrected before being used by eDMA\_A.

To prevent eDMA\_A's use of single-bit faults from the TCD RAM for eDMA\_A: Ensure that ERM\_CR0[ESCIE2] is set to 1 to enable the ERM to trigger an interrupt, and use the interrupt to manage properly any single-bit fault that occurs.

## 18.2 Introduction

### 18.2.1 Overview

The Error Reporting Module (ERM) provides information and optional interrupt notification on memory error events associated with ECC (Error Correction Code). The ERM collects ECC events on memory accesses for platform local memory arrays, such as flash memory, system RAM, or peripheral RAMs. See the chip-specific ERM information for details about supported memory sources and specific memory channel assignments.

## 18.2.2 Features

The ERM includes these features:

- Capture address and syndrome information on single-bit correction and non-correctable ECC events
- Optional interrupt notification on captured ECC events
- Support for ECC event capturing for memory sources, with individual reporting fields and interrupt configuration per memory channel.

## 18.3 Memory map and register definition

The ERM provides an IPS programming model mapped to an on-platform peripheral slot. The programming model can only be referenced using a 32-bit (word) access. Attempted references using different access sizes or in user mode generate an IPS error termination. The ERM allows access to the programming model by all system bus masters.

The programming model can only be accessed in supervisor mode.

Attempted accesses to undefined (reserved) memory regions can result in undefined behavior.

Attempted updates to the programming model while the ERM module is in the middle of an operation will result in non-deterministic behavior.

**ERM memory map**

| Address offset (hex) | Register name                                  | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|------------------------------------------------|-----------------|--------|-------------|----------------------------|
| 0                    | ERM Configuration Register (ERM_CR0)           | 32              | R/W    | 0000_0000h  | <a href="#">18.3.1/731</a> |
| 4                    | ERM Configuration Register (ERM_CR1)           | 32              | R/W    | 0000_0000h  | <a href="#">18.3.2/734</a> |
| 8                    | ERM Configuration Register (ERM_CR2)           | 32              | R/W    | 0000_0000h  | <a href="#">18.3.3/737</a> |
| 10                   | ERM Status Register (ERM_SR0)                  | 32              | w1c    | 0000_0000h  | <a href="#">18.3.4/740</a> |
| 14                   | ERM Status Register (ERM_SR1)                  | 32              | w1c    | 0000_0000h  | <a href="#">18.3.5/744</a> |
| 18                   | ERM Status Register (ERM_SR2)                  | 32              | w1c    | 0000_0000h  | <a href="#">18.3.6/748</a> |
| 100                  | ERM Memory n Error Address Register (ERM_EAR0) | 32              | R      | 0000_0000h  | <a href="#">18.3.7/750</a> |
| 104                  | ERM Memory n Syndrome Register (ERM_SYN0)      | 32              | R      | 0000_0000h  | <a href="#">18.3.8/751</a> |
| 110                  | ERM Memory n Error Address Register (ERM_EAR1) | 32              | R      | 0000_0000h  | <a href="#">18.3.7/750</a> |
| 114                  | ERM Memory n Syndrome Register (ERM_SYN1)      | 32              | R      | 0000_0000h  | <a href="#">18.3.8/751</a> |
| 120                  | ERM Memory n Error Address Register (ERM_EAR2) | 32              | R      | 0000_0000h  | <a href="#">18.3.7/750</a> |

*Table continues on the next page...*

## ERM memory map (continued)

| Address offset (hex) | Register name                                   | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|-------------------------------------------------|-----------------|--------|-------------|----------------------------|
| 124                  | ERM Memory n Syndrome Register (ERM_SYN2)       | 32              | R      | 0000_0000h  | <a href="#">18.3.8/751</a> |
| 130                  | ERM Memory n Error Address Register (ERM_EAR3)  | 32              | R      | 0000_0000h  | <a href="#">18.3.7/750</a> |
| 134                  | ERM Memory n Syndrome Register (ERM_SYN3)       | 32              | R      | 0000_0000h  | <a href="#">18.3.8/751</a> |
| 140                  | ERM Memory n Error Address Register (ERM_EAR4)  | 32              | R      | 0000_0000h  | <a href="#">18.3.7/750</a> |
| 144                  | ERM Memory n Syndrome Register (ERM_SYN4)       | 32              | R      | 0000_0000h  | <a href="#">18.3.8/751</a> |
| 150                  | ERM Memory n Error Address Register (ERM_EAR5)  | 32              | R      | 0000_0000h  | <a href="#">18.3.7/750</a> |
| 154                  | ERM Memory n Syndrome Register (ERM_SYN5)       | 32              | R      | 0000_0000h  | <a href="#">18.3.8/751</a> |
| 160                  | ERM Memory n Error Address Register (ERM_EAR6)  | 32              | R      | 0000_0000h  | <a href="#">18.3.7/750</a> |
| 164                  | ERM Memory n Syndrome Register (ERM_SYN6)       | 32              | R      | 0000_0000h  | <a href="#">18.3.8/751</a> |
| 170                  | ERM Memory n Error Address Register (ERM_EAR7)  | 32              | R      | 0000_0000h  | <a href="#">18.3.7/750</a> |
| 174                  | ERM Memory n Syndrome Register (ERM_SYN7)       | 32              | R      | 0000_0000h  | <a href="#">18.3.8/751</a> |
| 180                  | ERM Memory n Error Address Register (ERM_EAR8)  | 32              | R      | 0000_0000h  | <a href="#">18.3.7/750</a> |
| 184                  | ERM Memory n Syndrome Register (ERM_SYN8)       | 32              | R      | 0000_0000h  | <a href="#">18.3.8/751</a> |
| 190                  | ERM Memory n Error Address Register (ERM_EAR9)  | 32              | R      | 0000_0000h  | <a href="#">18.3.7/750</a> |
| 194                  | ERM Memory n Syndrome Register (ERM_SYN9)       | 32              | R      | 0000_0000h  | <a href="#">18.3.8/751</a> |
| 1A0                  | ERM Memory n Error Address Register (ERM_EAR10) | 32              | R      | 0000_0000h  | <a href="#">18.3.7/750</a> |
| 1A4                  | ERM Memory n Syndrome Register (ERM_SYN10)      | 32              | R      | 0000_0000h  | <a href="#">18.3.8/751</a> |
| 1B0                  | ERM Memory n Error Address Register (ERM_EAR11) | 32              | R      | 0000_0000h  | <a href="#">18.3.7/750</a> |
| 1B4                  | ERM Memory n Syndrome Register (ERM_SYN11)      | 32              | R      | 0000_0000h  | <a href="#">18.3.8/751</a> |
| 1C0                  | ERM Memory n Error Address Register (ERM_EAR12) | 32              | R      | 0000_0000h  | <a href="#">18.3.7/750</a> |
| 1C4                  | ERM Memory n Syndrome Register (ERM_SYN12)      | 32              | R      | 0000_0000h  | <a href="#">18.3.8/751</a> |
| 1D0                  | ERM Memory n Error Address Register (ERM_EAR13) | 32              | R      | 0000_0000h  | <a href="#">18.3.7/750</a> |
| 1D4                  | ERM Memory n Syndrome Register (ERM_SYN13)      | 32              | R      | 0000_0000h  | <a href="#">18.3.8/751</a> |
| 1E0                  | ERM Memory n Error Address Register (ERM_EAR14) | 32              | R      | 0000_0000h  | <a href="#">18.3.7/750</a> |
| 1E4                  | ERM Memory n Syndrome Register (ERM_SYN14)      | 32              | R      | 0000_0000h  | <a href="#">18.3.8/751</a> |
| 1F0                  | ERM Memory n Error Address Register (ERM_EAR15) | 32              | R      | 0000_0000h  | <a href="#">18.3.7/750</a> |
| 1F4                  | ERM Memory n Syndrome Register (ERM_SYN15)      | 32              | R      | 0000_0000h  | <a href="#">18.3.8/751</a> |
| 200                  | ERM Memory n Error Address Register (ERM_EAR16) | 32              | R      | 0000_0000h  | <a href="#">18.3.7/750</a> |
| 204                  | ERM Memory n Syndrome Register (ERM_SYN16)      | 32              | R      | 0000_0000h  | <a href="#">18.3.8/751</a> |
| 210                  | ERM Memory n Error Address Register (ERM_EAR17) | 32              | R      | 0000_0000h  | <a href="#">18.3.7/750</a> |
| 214                  | ERM Memory n Syndrome Register (ERM_SYN17)      | 32              | R      | 0000_0000h  | <a href="#">18.3.8/751</a> |
| 220                  | ERM Memory n Error Address Register (ERM_EAR18) | 32              | R      | 0000_0000h  | <a href="#">18.3.7/750</a> |
| 224                  | ERM Memory n Syndrome Register (ERM_SYN18)      | 32              | R      | 0000_0000h  | <a href="#">18.3.8/751</a> |
| 230                  | ERM Memory n Error Address Register (ERM_EAR19) | 32              | R      | 0000_0000h  | <a href="#">18.3.7/750</a> |
| 234                  | ERM Memory n Syndrome Register (ERM_SYN19)      | 32              | R      | 0000_0000h  | <a href="#">18.3.8/751</a> |

### 18.3.1 ERM Configuration Register (ERM\_CR0)

The ERM Configuration Register is a 32-bit control register for configuring the interrupt notification capability for memory channels 0 to 7.

Address: 0h base + 0h offset = 0h

|       |        |        |   |   |        |        |   |   |        |        |    |    |        |        |    |    |
|-------|--------|--------|---|---|--------|--------|---|---|--------|--------|----|----|--------|--------|----|----|
| Bit   | 0      | 1      | 2 | 3 | 4      | 5      | 6 | 7 | 8      | 9      | 10 | 11 | 12     | 13     | 14 | 15 |
| R     | ESCIE0 | ENCIE0 | 0 |   | ESCIE1 | ENCIE1 | 0 |   | ESCIE2 | ENCIE2 | 0  |    | ESCIE3 | ENCIE3 | 0  |    |
| W     |        |        |   |   |        |        |   |   |        |        |    |    |        |        |    |    |
| Reset | 0      | 0      | 0 | 0 | 0      | 0      | 0 | 0 | 0      | 0      | 0  | 0  | 0      | 0      | 0  | 0  |

|       |        |        |    |    |        |        |    |    |        |        |    |    |        |        |    |    |
|-------|--------|--------|----|----|--------|--------|----|----|--------|--------|----|----|--------|--------|----|----|
| Bit   | 16     | 17     | 18 | 19 | 20     | 21     | 22 | 23 | 24     | 25     | 26 | 27 | 28     | 29     | 30 | 31 |
| R     | ESCIE4 | ENCIE4 | 0  |    | ESCIE5 | ENCIE5 | 0  |    | ESCIE6 | ENCIE6 | 0  |    | ESCIE7 | ENCIE7 | 0  |    |
| W     |        |        |    |    |        |        |    |    |        |        |    |    |        |        |    |    |
| Reset | 0      | 0      | 0  | 0  | 0      | 0      | 0  | 0  | 0      | 0      | 0  | 0  | 0      | 0      | 0  | 0  |

**ERM\_CR0 field descriptions**

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                    |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>ESCIE0     | <p>Enable Memory 0 Single Correction Interrupt Notification</p> <p>This field is initialized by hardware reset.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 0 mapping.</p> <p>0 Interrupt notification of Memory 0 single-bit correction events is disabled.</p> <p>1 Interrupt notification of Memory 0 single-bit correction events is enabled.</p> |
| 1<br>ENCIE0     | <p>Enable Memory 0 Non-Correctable Interrupt Notification</p> <p>This field is initialized by hardware reset.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 0 mapping.</p> <p>0 Interrupt notification of Memory 0 non-correctable error events is disabled.</p> <p>1 Interrupt notification of Memory 0 non-correctable error events is enabled.</p>   |
| 2–3<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                             |
| 4<br>ESCIE1     | <p>Enable Memory 1 Single Correction Interrupt Notification</p> <p>This field is initialized by hardware reset.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 1 mapping.</p> <p>0 Interrupt notification of Memory 1 single-bit correction events is disabled.</p> <p>1 Interrupt notification of Memory 1 single-bit correction events is enabled.</p> |

*Table continues on the next page...*

## ERM\_CR0 field descriptions (continued)

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                 |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 5<br>ENCIE1       | <p>Enable Memory 1 Non-Correctable Interrupt Notification</p> <p>This field is initialized by hardware reset.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 1 mapping.</p> <p>0 Interrupt notification of Memory 1 non-correctable error events is disabled.<br/>1 Interrupt notification of Memory 1 non-correctable error events is enabled.</p>   |
| 6–7<br>Reserved   | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                          |
| 8<br>ESCIE2       | <p>Enable Memory 2 Single Correction Interrupt Notification</p> <p>This field is initialized by hardware reset.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 2 mapping.</p> <p>0 Interrupt notification of Memory 2 single-bit correction events is disabled.<br/>1 Interrupt notification of Memory 2 single-bit correction events is enabled.</p> |
| 9<br>ENCIE2       | <p>Enable Memory 2 Non-Correctable Interrupt Notification</p> <p>This field is initialized by hardware reset.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 2 mapping.</p> <p>0 Interrupt notification of Memory 2 non-correctable error events is disabled.<br/>1 Interrupt notification of Memory 2 non-correctable error events is enabled.</p>   |
| 10–11<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                          |
| 12<br>ESCIE3      | <p>Enable Memory 3 Single Correction Interrupt Notification</p> <p>This field is initialized by hardware reset.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 3 mapping.</p> <p>0 Interrupt notification of Memory 3 single-bit correction events is disabled.<br/>1 Interrupt notification of Memory 3 single-bit correction events is enabled.</p> |
| 13<br>ENCIE3      | <p>Enable Memory 3 Non-Correctable Interrupt Notification</p> <p>This field is initialized by hardware reset.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 3 mapping.</p> <p>0 Interrupt notification of Memory 3 non-correctable error events is disabled.<br/>1 Interrupt notification of Memory 3 non-correctable error events is enabled.</p>   |
| 14–15<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                          |
| 16<br>ESCIE4      | <p>Enable Memory 4 Single Correction Interrupt Notification</p> <p>This field is initialized by hardware reset.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 4 mapping.</p> <p>0 Interrupt notification of Memory 4 single-bit correction events is disabled.<br/>1 Interrupt notification of Memory 4 single-bit correction events is enabled.</p> |
| 17<br>ENCIE4      | <p>Enable Memory 4 Non-Correctable Interrupt Notification</p>                                                                                                                                                                                                                                                                                                                               |

*Table continues on the next page...*



## ERM\_CR0 field descriptions (continued)

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | <p>This field is initialized by hardware reset.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 4 mapping.</p> <p>0 Interrupt notification of Memory 4 non-correctable error events is disabled.</p> <p>1 Interrupt notification of Memory 4 non-correctable error events is enabled.</p>                                                                 |
| 18–19<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                             |
| 20<br>ESCIE5      | <p>Enable Memory 5 Single Correction Interrupt Notification</p> <p>This field is initialized by hardware reset.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 5 mapping.</p> <p>0 Interrupt notification of Memory 5 single-bit correction events is disabled.</p> <p>1 Interrupt notification of Memory 5 single-bit correction events is enabled.</p> |
| 21<br>ENCIE5      | <p>Enable Memory 5 Non-Correctable Interrupt Notification</p> <p>This field is initialized by hardware reset.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 5 mapping.</p> <p>0 Interrupt notification of Memory 5 non-correctable error events is disabled.</p> <p>1 Interrupt notification of Memory 5 non-correctable error events is enabled.</p>   |
| 22–23<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                             |
| 24<br>ESCIE6      | <p>Enable Memory 6 Single Correction Interrupt Notification</p> <p>This field is initialized by hardware reset.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 6 mapping.</p> <p>0 Interrupt notification of Memory 6 single-bit correction events is disabled.</p> <p>1 Interrupt notification of Memory 6 single-bit correction events is enabled.</p> |
| 25<br>ENCIE6      | <p>Enable Memory 6 Non-Correctable Interrupt Notification</p> <p>This field is initialized by hardware reset.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 6 mapping.</p> <p>0 Interrupt notification of Memory 6 non-correctable error events is disabled.</p> <p>1 Interrupt notification of Memory 6 non-correctable error events is enabled.</p>   |
| 26–27<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                             |
| 28<br>ESCIE7      | <p>Enable Memory 7 Single Correction Interrupt Notification</p> <p>This field is initialized by hardware reset.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 7 mapping.</p> <p>0 Interrupt notification of Memory 7 single-bit correction events is disabled.</p> <p>1 Interrupt notification of Memory 7 single-bit correction events is enabled.</p> |
| 29<br>ENCIE7      | <p>Enable Memory 7 Non-Correctable Interrupt Notification</p> <p>This field is initialized by hardware reset.</p>                                                                                                                                                                                                                                                                              |

*Table continues on the next page...*

## ERM\_CR0 field descriptions (continued)

| Field             | Description                                                                                                                                                                                                                                                 |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | <b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 7 mapping.<br>0 Interrupt notification of Memory 7 non-correctable error events is disabled.<br>1 Interrupt notification of Memory 7 non-correctable error events is enabled. |
| 30–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                     |

## 18.3.2 ERM Configuration Register (ERM\_CR1)

The ERM Configuration Register is a 32-bit control register for configuring the interrupt notification capability for memory channels 8 to 15.

Address: 0h base + 4h offset = 4h

|       |        |        |   |   |        |        |   |   |         |         |    |    |         |         |    |    |
|-------|--------|--------|---|---|--------|--------|---|---|---------|---------|----|----|---------|---------|----|----|
| Bit   | 0      | 1      | 2 | 3 | 4      | 5      | 6 | 7 | 8       | 9       | 10 | 11 | 12      | 13      | 14 | 15 |
| R     | ESCIE8 | ENCIE8 | 0 |   | ESCIE9 | ENCIE9 | 0 |   | ESCIE10 | ENCIE10 | 0  |    | ESCIE11 | ENCIE11 | 0  |    |
| W     | ESCIE8 | ENCIE8 |   |   | ESCIE9 | ENCIE9 |   |   | ESCIE10 | ENCIE10 |    |    | ESCIE11 | ENCIE11 |    |    |
| Reset | 0      | 0      | 0 | 0 | 0      | 0      | 0 | 0 | 0       | 0       | 0  | 0  | 0       | 0       | 0  | 0  |

|       |         |         |    |    |         |         |    |    |         |         |    |    |         |         |    |    |
|-------|---------|---------|----|----|---------|---------|----|----|---------|---------|----|----|---------|---------|----|----|
| Bit   | 16      | 17      | 18 | 19 | 20      | 21      | 22 | 23 | 24      | 25      | 26 | 27 | 28      | 29      | 30 | 31 |
| R     | ESCIE12 | ENCIE12 | 0  |    | ESCIE13 | ENCIE13 | 0  |    | ESCIE14 | ENCIE14 | 0  |    | ESCIE15 | ENCIE15 | 0  |    |
| W     | ESCIE12 | ENCIE12 |    |    | ESCIE13 | ENCIE13 |    |    | ESCIE14 | ENCIE14 |    |    | ESCIE15 | ENCIE15 |    |    |
| Reset | 0       | 0       | 0  | 0  | 0       | 0       | 0  | 0  | 0       | 0       | 0  | 0  | 0       | 0       | 0  | 0  |

## ERM\_CR1 field descriptions

| Field       | Description                                                                                                                                                                                                                                                                                                                                                             |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>ESCIE8 | Enable Memory 8 Single Correction Interrupt Notification<br>This field is initialized by hardware reset.<br><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 8 mapping.<br>0 Interrupt notification of Memory 8 single-bit correction events is disabled.<br>1 Interrupt notification of Memory 8 single-bit correction events is enabled. |
| 1<br>ENCIE8 | Enable Memory 8 Non-Correctable Interrupt Notification<br>This field is initialized by hardware reset.<br><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 8 mapping.                                                                                                                                                                      |

Table continues on the next page...

**ERM\_CR1 field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                 |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | 0 Interrupt notification of Memory 8 non-correctable error events is disabled.<br>1 Interrupt notification of Memory 8 non-correctable error events is enabled.                                                                                                                                                                                                             |
| 2–3<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                     |
| 4<br>ESCIE9       | Enable Memory 9 Single Correction Interrupt Notification<br>This field is initialized by hardware reset.<br><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 9 mapping.<br>0 Interrupt notification of Memory 9 single-bit correction events is disabled.<br>1 Interrupt notification of Memory 9 single-bit correction events is enabled.     |
| 5<br>ENCIE9       | Enable Memory 9 Non-Correctable Interrupt Notification<br>This field is initialized by hardware reset.<br><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 9 mapping.<br>0 Interrupt notification of Memory 9 non-correctable error events is disabled.<br>1 Interrupt notification of Memory 9 non-correctable error events is enabled.       |
| 6–7<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                     |
| 8<br>ESCIE10      | Enable Memory 10 Single Correction Interrupt Notification<br>This field is initialized by hardware reset.<br><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 10 mapping.<br>0 Interrupt notification of Memory 10 single-bit correction events is disabled.<br>1 Interrupt notification of Memory 10 single-bit correction events is enabled. |
| 9<br>ENCIE10      | Enable Memory 10 Non-Correctable Interrupt Notification<br>This field is initialized by hardware reset.<br><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 10 mapping.<br>0 Interrupt notification of Memory 10 non-correctable error events is disabled.<br>1 Interrupt notification of Memory 10 non-correctable error events is enabled.   |
| 10–11<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                     |
| 12<br>ESCIE11     | Enable Memory 11 Single Correction Interrupt Notification<br>This field is initialized by hardware reset.<br><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 11 mapping.<br>0 Interrupt notification of Memory 11 single-bit correction events is disabled.<br>1 Interrupt notification of Memory 11 single-bit correction events is enabled. |
| 13<br>ENCIE11     | Enable Memory 11 Non-Correctable Interrupt Notification<br>This field is initialized by hardware reset.<br><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 11 mapping.<br>0 Interrupt notification of Memory 11 non-correctable error events is disabled.<br>1 Interrupt notification of Memory 11 non-correctable error events is enabled.   |

*Table continues on the next page...*

## ERM\_CR1 field descriptions (continued)

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                 |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 14–15<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                     |
| 16<br>ESCIE12     | Enable Memory 12 Single Correction Interrupt Notification<br>This field is initialized by hardware reset.<br><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 12 mapping.<br>0 Interrupt notification of Memory 12 single-bit correction events is disabled.<br>1 Interrupt notification of Memory 12 single-bit correction events is enabled. |
| 17<br>ENCIE12     | Enable Memory 12 Non-Correctable Interrupt Notification<br>This field is initialized by hardware reset.<br><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 12 mapping.<br>0 Interrupt notification of Memory 12 non-correctable error events is disabled.<br>1 Interrupt notification of Memory 12 non-correctable error events is enabled.   |
| 18–19<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                     |
| 20<br>ESCIE13     | Enable Memory 13 Single Correction Interrupt Notification<br>This field is initialized by hardware reset.<br><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 13 mapping.<br>0 Interrupt notification of Memory 13 single-bit correction events is disabled.<br>1 Interrupt notification of Memory 13 single-bit correction events is enabled. |
| 21<br>ENCIE13     | Enable Memory 13 Non-Correctable Interrupt Notification<br>This field is initialized by hardware reset.<br><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 13 mapping.<br>0 Interrupt notification of Memory 13 non-correctable error events is disabled.<br>1 Interrupt notification of Memory 13 non-correctable error events is enabled.   |
| 22–23<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                     |
| 24<br>ESCIE14     | Enable Memory 14 Single Correction Interrupt Notification<br>This field is initialized by hardware reset.<br><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 14 mapping.<br>0 Interrupt notification of Memory 14 single-bit correction events is disabled.<br>1 Interrupt notification of Memory 14 single-bit correction events is enabled. |
| 25<br>ENCIE14     | Enable Memory 14 Non-Correctable Interrupt Notification<br>This field is initialized by hardware reset.<br><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 14 mapping.<br>0 Interrupt notification of Memory 14 non-correctable error events is disabled.<br>1 Interrupt notification of Memory 14 non-correctable error events is enabled.   |
| 26–27<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                     |

Table continues on the next page...

**ERM\_CR1 field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 28<br>ESCIE15     | <p>Enable Memory 15 Single Correction Interrupt Notification</p> <p>This field is initialized by hardware reset.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 15 mapping.</p> <p>0 Interrupt notification of Memory 15 single-bit correction events is disabled.<br/>1 Interrupt notification of Memory 15 single-bit correction events is enabled.</p> |
| 29<br>ENCIE15     | <p>Enable Memory 15 Non-Correctable Interrupt Notification</p> <p>This field is initialized by hardware reset.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 15 mapping.</p> <p>0 Interrupt notification of Memory 15 non-correctable error events is disabled.<br/>1 Interrupt notification of Memory 15 non-correctable error events is enabled.</p>   |
| 30–31<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                              |

**18.3.3 ERM Configuration Register (ERM\_CR2)**

The ERM Configuration Register is a 32-bit control register for configuring the interrupt notification capability for memory channels 16 to 19.

Address: 0h base + 8h offset = 8h

|       |         |         |   |   |         |         |   |   |         |         |    |    |         |         |    |    |
|-------|---------|---------|---|---|---------|---------|---|---|---------|---------|----|----|---------|---------|----|----|
| Bit   | 0       | 1       | 2 | 3 | 4       | 5       | 6 | 7 | 8       | 9       | 10 | 11 | 12      | 13      | 14 | 15 |
| R     | ESCIE16 | ENCIE16 | 0 |   | ESCIE17 | ENCIE17 | 0 |   | ESCIE18 | ENCIE18 | 0  |    | ESCIE19 | ENCIE19 | 0  |    |
| W     |         |         |   |   |         |         |   |   |         |         |    |    |         |         |    |    |
| Reset | 0       | 0       | 0 | 0 | 0       | 0       | 0 | 0 | 0       | 0       | 0  | 0  | 0       | 0       | 0  | 0  |

|       |          |    |    |    |          |    |    |    |          |    |    |    |          |    |    |    |
|-------|----------|----|----|----|----------|----|----|----|----------|----|----|----|----------|----|----|----|
| Bit   | 16       | 17 | 18 | 19 | 20       | 21 | 22 | 23 | 24       | 25 | 26 | 27 | 28       | 29 | 30 | 31 |
| R     | Reserved | 0  |    |    | Reserved |    | 0  |    | Reserved |    | 0  |    | Reserved |    | 0  |    |
| W     |          |    |    |    |          |    |    |    |          |    |    |    |          |    |    |    |
| Reset | 0        | 0  | 0  | 0  | 0        | 0  | 0  | 0  | 0        | 0  | 0  | 0  | 0        | 0  | 0  | 0  |

**ERM\_CR2 field descriptions**

| Field        | Description                                                                                                                                                                                                           |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>ESCIE16 | <p>Enable Memory 16 Single Correction Interrupt Notification</p> <p>This field is initialized by hardware reset.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 16 mapping.</p> |

*Table continues on the next page...*

## ERM\_CR2 field descriptions (continued)

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                 |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | 0 Interrupt notification of Memory 16 single-bit correction events is disabled.<br>1 Interrupt notification of Memory 16 single-bit correction events is enabled.                                                                                                                                                                                                           |
| 1<br>ENCIE16      | Enable Memory 16 Non-Correctable Interrupt Notification<br>This field is initialized by hardware reset.<br><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 16 mapping.<br>0 Interrupt notification of Memory 16 non-correctable error events is disabled.<br>1 Interrupt notification of Memory 16 non-correctable error events is enabled.   |
| 2–3<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                     |
| 4<br>ESCIE17      | Enable Memory 17 Single Correction Interrupt Notification<br>This field is initialized by hardware reset.<br><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 17 mapping.<br>0 Interrupt notification of Memory 17 single-bit correction events is disabled.<br>1 Interrupt notification of Memory 17 single-bit correction events is enabled. |
| 5<br>ENCIE17      | Enable Memory 17 Non-Correctable Interrupt Notification<br>This field is initialized by hardware reset.<br><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 17 mapping.<br>0 Interrupt notification of Memory 17 non-correctable error events is disabled.<br>1 Interrupt notification of Memory 17 non-correctable error events is enabled.   |
| 6–7<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                     |
| 8<br>ESCIE18      | Enable Memory 18 Single Correction Interrupt Notification<br>This field is initialized by hardware reset.<br><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 18 mapping.<br>0 Interrupt notification of Memory 18 single-bit correction events is disabled.<br>1 Interrupt notification of Memory 18 single-bit correction events is enabled. |
| 9<br>ENCIE18      | Enable Memory 18 Non-Correctable Interrupt Notification<br>This field is initialized by hardware reset.<br><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 18 mapping.<br>0 Interrupt notification of Memory 18 non-correctable error events is disabled.<br>1 Interrupt notification of Memory 18 non-correctable error events is enabled.   |
| 10–11<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                     |
| 12<br>ESCIE19     | Enable Memory 19 Single Correction Interrupt Notification<br>This field is initialized by hardware reset.<br><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 19 mapping.<br>0 Interrupt notification of Memory 19 single-bit correction events is disabled.<br>1 Interrupt notification of Memory 19 single-bit correction events is enabled. |

Table continues on the next page...

**ERM\_CR2 field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 13<br>ENCIE19     | <p>Enable Memory 19 Non-Correctable Interrupt Notification</p> <p>This field is initialized by hardware reset.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 19 mapping.</p> <p>0 Interrupt notification of Memory 19 non-correctable error events is disabled.<br/>1 Interrupt notification of Memory 19 non-correctable error events is enabled.</p> |
| 14–15<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                            |
| 16–17<br>Reserved | <p>This field is reserved.</p>                                                                                                                                                                                                                                                                                                                                                                |
| 18–19<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                            |
| 20–21<br>Reserved | <p>This field is reserved.</p>                                                                                                                                                                                                                                                                                                                                                                |
| 22–23<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                            |
| 24–25<br>Reserved | <p>This field is reserved.</p>                                                                                                                                                                                                                                                                                                                                                                |
| 26–27<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                            |
| 28–29<br>Reserved | <p>This field is reserved.</p>                                                                                                                                                                                                                                                                                                                                                                |
| 30–31<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                            |

### 18.3.4 ERM Status Register (ERM\_SR0)

The ERM Status Register is a 32-bit control register for signaling which types of ECC events have been detected for each memory channel. The register signals the last memory event to be detected for memory channels 0 to 7.

Address: 0h base + 10h offset = 10h

| Bit   | 0    | 1    | 2  | 3  | 4    | 5    | 6  | 7  | 8    | 9    | 10 | 11 | 12   | 13   | 14 | 15 |
|-------|------|------|----|----|------|------|----|----|------|------|----|----|------|------|----|----|
| R     | SBC0 | NCE0 | 0  |    | SBC1 | NCE1 | 0  |    | SBC2 | NCE2 | 0  |    | SBC3 | NCE3 | 0  |    |
| W     | w1c  | w1c  |    |    | w1c  | w1c  |    |    | w1c  | w1c  |    |    | w1c  | w1c  |    |    |
| Reset | 0    | 0    | 0  | 0  | 0    | 0    | 0  | 0  | 0    | 0    | 0  | 0  | 0    | 0    | 0  | 0  |
| Bit   | 16   | 17   | 18 | 19 | 20   | 21   | 22 | 23 | 24   | 25   | 26 | 27 | 28   | 29   | 30 | 31 |
| R     | SBC4 | NCE4 | 0  |    | SBC5 | NCE5 | 0  |    | SBC6 | NCE6 | 0  |    | SBC7 | NCE7 | 0  |    |
| W     | w1c  | w1c  |    |    | w1c  | w1c  |    |    | w1c  | w1c  |    |    | w1c  | w1c  |    |    |
| Reset | 0    | 0    | 0  | 0  | 0    | 0    | 0  | 0  | 0    | 0    | 0  | 0  | 0    | 0    | 0  | 0  |

**ERM\_SR0 field descriptions**

| Field     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>SBC0 | <p>Memory 0 Single-Bit Correction Event</p> <p>This field is initialized by hardware reset.</p> <p>Write 1 to clear this field. This write also clears the corresponding interrupt notification if ERM_CR0[ESCIE0] is enabled.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 0 mapping.</p> <p>0 No single-bit correction event on Memory 0 detected<br/>1 Single-bit correction event on Memory 0 detected</p> |
| 1<br>NCE0 | <p>Memory 0 Non-Correctable Error Event</p> <p>This field is initialized by hardware reset.</p> <p>Write 1 to clear this field. This write also clears the corresponding interrupt notification if ERM_CR0[ENCIE0] is enabled.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 0 mapping.</p>                                                                                                                     |

*Table continues on the next page...*



## ERM\_SR0 field descriptions (continued)

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | 0 No non-correctable error event on Memory 0 detected<br>1 Non-correctable error event on Memory 0 detected                                                                                                                                                                                                                                                                                                                    |
| 2–3<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                        |
| 4<br>SBC1         | Memory 1 Single-Bit Correction Event<br>This field is initialized by hardware reset.<br>Write 1 to clear this field. This write also clears the corresponding interrupt notification if ERM_CR0[ESCIE1] is enabled.<br><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 1 mapping.<br>0 No single-bit correction event on Memory 1 detected<br>1 Single-bit correction event on Memory 1 detected |
| 5<br>NCE1         | Memory 1 Non-Correctable Error Event<br>This field is initialized by hardware reset.<br>Write 1 to clear this field. This write also clears the corresponding interrupt notification if ERM_CR0[ENCIE1] is enabled.<br><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 1 mapping.<br>0 No non-correctable error event on Memory 1 detected<br>1 Non-correctable error event on Memory 1 detected |
| 6–7<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                        |
| 8<br>SBC2         | Memory 2 Single-Bit Correction Event<br>This field is initialized by hardware reset.<br>Write 1 to clear this field. This write also clears the corresponding interrupt notification if ERM_CR0[ESCIE2] is enabled.<br><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 2 mapping.<br>0 No single-bit correction event on Memory 2 detected<br>1 Single-bit correction event on Memory 2 detected |
| 9<br>NCE2         | Memory 2 Non-Correctable Error Event<br>This field is initialized by hardware reset.<br>Write 1 to clear this field. This write also clears the corresponding interrupt notification if ERM_CR0[ENCIE2] is enabled.<br><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 2 mapping.<br>0 No non-correctable error event on Memory 2 detected<br>1 Non-correctable error event on Memory 2 detected |
| 10–11<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                        |
| 12<br>SBC3        | Memory 3 Single-Bit Correction Event<br>This field is initialized by hardware reset.<br>Write 1 to clear this field. This write also clears the corresponding interrupt notification if ERM_CR0[ESCIE3] is enabled.                                                                                                                                                                                                            |

*Table continues on the next page...*

## ERM\_SR0 field descriptions (continued)

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 3 mapping.</p> <p>0 No single-bit correction event on Memory 3 detected</p> <p>1 Single-bit correction event on Memory 3 detected</p>                                                                                                                                                                                                                                    |
| 13<br>NCE3        | <p>Memory 3 Non-Correctable Error Event</p> <p>This field is initialized by hardware reset.</p> <p>Write 1 to clear this field. This write also clears the corresponding interrupt notification if ERM_CR0[ENCIE3] is enabled.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 3 mapping.</p> <p>0 No non-correctable error event on Memory 3 detected</p> <p>1 Non-correctable error event on Memory 3 detected</p> |
| 14–15<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                        |
| 16<br>SBC4        | <p>Memory 4 Single-Bit Correction Event</p> <p>This field is initialized by hardware reset.</p> <p>Write 1 to clear this field. This write also clears the corresponding interrupt notification if ERM_CR0[ESCIE4] is enabled.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 4 mapping.</p> <p>0 No single-bit correction event on Memory 4 detected</p> <p>1 Single-bit correction event on Memory 4 detected</p> |
| 17<br>NCE4        | <p>Memory 4 Non-Correctable Error Event</p> <p>This field is initialized by hardware reset.</p> <p>Write 1 to clear this field. This write also clears the corresponding interrupt notification if ERM_CR0[ENCIE4] is enabled.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 4 mapping.</p> <p>0 No non-correctable error event on Memory 4 detected</p> <p>1 Non-correctable error event on Memory 4 detected</p> |
| 18–19<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                        |
| 20<br>SBC5        | <p>Memory 5 Single-Bit Correction Event</p> <p>This field is initialized by hardware reset.</p> <p>Write 1 to clear this field. This write also clears the corresponding interrupt notification if ERM_CR0[ESCIE5] is enabled.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 5 mapping.</p> <p>0 No single-bit correction event on Memory 5 detected</p> <p>1 Single-bit correction event on Memory 5 detected</p> |
| 21<br>NCE5        | <p>Memory 5 Non-Correctable Error Event</p> <p>This field is initialized by hardware reset.</p> <p>Write 1 to clear this field. This write also clears the corresponding interrupt notification if ERM_CR0[ENCIE5] is enabled.</p>                                                                                                                                                                                                                        |

*Table continues on the next page...*

## ERM\_SR0 field descriptions (continued)

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 5 mapping.</p> <p>0 No non-correctable error event on Memory 5 detected</p> <p>1 Non-correctable error event on Memory 5 detected</p>                                                                                                                                                                                                                                    |
| 22–23<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                        |
| 24<br>SBC6        | <p>Memory 6 Single-Bit Correction Event</p> <p>This field is initialized by hardware reset.</p> <p>Write 1 to clear this field. This write also clears the corresponding interrupt notification if ERM_CR0[ESCIE6] is enabled.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 6 mapping.</p> <p>0 No single-bit correction event on Memory 6 detected</p> <p>1 Single-bit correction event on Memory 6 detected</p> |
| 25<br>NCE6        | <p>Memory 6 Non-Correctable Error Event</p> <p>This field is initialized by hardware reset.</p> <p>Write 1 to clear this field. This write also clears the corresponding interrupt notification if ERM_CR0[ENCIE6] is enabled.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 6 mapping.</p> <p>0 No non-correctable error event on Memory 6 detected</p> <p>1 Non-correctable error event on Memory 6 detected</p> |
| 26–27<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                        |
| 28<br>SBC7        | <p>Memory 7 Single-Bit Correction Event</p> <p>This field is initialized by hardware reset.</p> <p>Write 1 to clear this field. This write also clears the corresponding interrupt notification if ERM_CR0[ESCIE7] is enabled.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 7 mapping.</p> <p>0 No single-bit correction event on Memory 7 detected</p> <p>1 Single-bit correction event on Memory 7 detected</p> |
| 29<br>NCE7        | <p>Memory 7 Non-Correctable Error Event</p> <p>This field is initialized by hardware reset.</p> <p>Write 1 to clear this field. This write also clears the corresponding interrupt notification if ERM_CR0[ENCIE7] is enabled.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 7 mapping.</p> <p>0 No non-correctable error event on Memory 7 detected</p> <p>1 Non-correctable error event on Memory 7 detected</p> |
| 30–31<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                        |

### 18.3.5 ERM Status Register (ERM\_SR1)

The ERM Status Register is a 32-bit control register for signaling which types of ECC events have been detected for each memory channel. The register signals the last memory event to be detected for memory channels 8 to 15.

Address: 0h base + 14h offset = 14h

|       |      |      |   |   |      |      |   |   |       |       |    |    |       |       |    |    |
|-------|------|------|---|---|------|------|---|---|-------|-------|----|----|-------|-------|----|----|
| Bit   | 0    | 1    | 2 | 3 | 4    | 5    | 6 | 7 | 8     | 9     | 10 | 11 | 12    | 13    | 14 | 15 |
| R     | SBC8 | NCE8 | 0 |   | SBC9 | NCE9 | 0 |   | SBC10 | NCE10 | 0  |    | SBC11 | NCE11 | 0  |    |
| W     | w1c  | w1c  |   |   | w1c  | w1c  |   |   | w1c   | w1c   |    |    | w1c   | w1c   |    |    |
| Reset | 0    | 0    | 0 | 0 | 0    | 0    | 0 | 0 | 0     | 0     | 0  | 0  | 0     | 0     | 0  | 0  |

|       |       |       |    |    |       |       |    |    |       |       |    |    |       |       |    |    |
|-------|-------|-------|----|----|-------|-------|----|----|-------|-------|----|----|-------|-------|----|----|
| Bit   | 16    | 17    | 18 | 19 | 20    | 21    | 22 | 23 | 24    | 25    | 26 | 27 | 28    | 29    | 30 | 31 |
| R     | SBC12 | NCE12 | 0  |    | SBC13 | NCE13 | 0  |    | SBC14 | NCE14 | 0  |    | SBC15 | NCE15 | 0  |    |
| W     | w1c   | w1c   |    |    | w1c   | w1c   |    |    | w1c   | w1c   |    |    | w1c   | w1c   |    |    |
| Reset | 0     | 0     | 0  | 0  | 0     | 0     | 0  | 0  | 0     | 0     | 0  | 0  | 0     | 0     | 0  | 0  |

**ERM\_SR1 field descriptions**

| Field     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>SBC8 | <p>Memory 8 Single-Bit Correction Event</p> <p>This field is initialized by hardware reset.</p> <p>Write 1 to clear this field. This write also clears the corresponding interrupt notification if ERM_CR1[ESCIE8] is enabled.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 8 mapping.</p> <p>0 No single-bit correction event on Memory 8 detected<br/>1 Single-bit correction event on Memory 8 detected</p> |
| 1<br>NCE8 | <p>Memory 8 Non-Correctable Error Event</p> <p>This field is initialized by hardware reset.</p> <p>Write 1 to clear this field. This write also clears the corresponding interrupt notification if ERM_CR1[ENCIE8] is enabled.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 8 mapping.</p>                                                                                                                     |

*Table continues on the next page...*

## ERM\_SR1 field descriptions (continued)

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | 0 No non-correctable error event on Memory 8 detected<br>1 Non-correctable error event on Memory 8 detected                                                                                                                                                                                                                                                                                                                         |
| 2–3<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                             |
| 4<br>SBC9         | Memory 9 Single-Bit Correction Event<br>This field is initialized by hardware reset.<br>Write 1 to clear this field. This write also clears the corresponding interrupt notification if ERM_CR1[ESCIE9] is enabled.<br><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 9 mapping.<br>0 No single-bit correction event on Memory 9 detected<br>1 Single-bit correction event on Memory 9 detected      |
| 5<br>NCE9         | Memory 9 Non-Correctable Error Event<br>This field is initialized by hardware reset.<br>Write 1 to clear this field. This write also clears the corresponding interrupt notification if ERM_CR1[ENCIE9] is enabled.<br><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 9 mapping.<br>0 No non-correctable error event on Memory 9 detected<br>1 Non-correctable error event on Memory 9 detected      |
| 6–7<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                             |
| 8<br>SBC10        | Memory 10 Single-Bit Correction Event<br>This field is initialized by hardware reset.<br>Write 1 to clear this field. This write also clears the corresponding interrupt notification if ERM_CR1[ESCIE10] is enabled.<br><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 10 mapping.<br>0 No single-bit correction event on Memory 10 detected<br>1 Single-bit correction event on Memory 10 detected |
| 9<br>NCE10        | Memory 10 Non-Correctable Error Event<br>This field is initialized by hardware reset.<br>Write 1 to clear this field. This write also clears the corresponding interrupt notification if ERM_CR1[ENCIE10] is enabled.<br><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 10 mapping.<br>0 No non-correctable error event on Memory 10 detected<br>1 Non-correctable error event on Memory 10 detected |
| 10–11<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                             |
| 12<br>SBC11       | Memory 11 Single-Bit Correction Event<br>This field is initialized by hardware reset.<br>Write 1 to clear this field. This write also clears the corresponding interrupt notification if ERM_CR1[ESCIE11] is enabled.                                                                                                                                                                                                               |

*Table continues on the next page...*

**ERM\_SR1 field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 11 mapping.</p> <p>0 No single-bit correction event on Memory 11 detected</p> <p>1 Single-bit correction event on Memory 11 detected</p>                                                                                                                                                                                                                                      |
| 13<br>NCE11       | <p>Memory 11 Non-Correctable Error Event</p> <p>This field is initialized by hardware reset.</p> <p>Write 1 to clear this field. This write also clears the corresponding interrupt notification if ERM_CR1[ENCIE11] is enabled.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 11 mapping.</p> <p>0 No non-correctable error event on Memory 11 detected</p> <p>1 Non-correctable error event on Memory 11 detected</p> |
| 14–15<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                             |
| 16<br>SBC12       | <p>Memory 12 Single-Bit Correction Event</p> <p>This field is initialized by hardware reset.</p> <p>Write 1 to clear this field. This write also clears the corresponding interrupt notification if ERM_CR1[ESCIE12] is enabled.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 12 mapping.</p> <p>0 No single-bit correction event on Memory 12 detected</p> <p>1 Single-bit correction event on Memory 12 detected</p> |
| 17<br>NCE12       | <p>Memory 12 Non-Correctable Error Event</p> <p>This field is initialized by hardware reset.</p> <p>Write 1 to clear this field. This write also clears the corresponding interrupt notification if ERM_CR1[ENCIE12] is enabled.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 12 mapping.</p> <p>0 No non-correctable error event on Memory 12 detected</p> <p>1 Non-correctable error event on Memory 12 detected</p> |
| 18–19<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                             |
| 20<br>SBC13       | <p>Memory 13 Single-Bit Correction Event</p> <p>This field is initialized by hardware reset.</p> <p>Write 1 to clear this field. This write also clears the corresponding interrupt notification if ERM_CR1[ESCIE13] is enabled.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 13 mapping.</p> <p>0 No single-bit correction event on Memory 13 detected</p> <p>1 Single-bit correction event on Memory 13 detected</p> |
| 21<br>NCE13       | <p>Memory 13 Non-Correctable Error Event</p> <p>This field is initialized by hardware reset.</p> <p>Write 1 to clear this field. This write also clears the corresponding interrupt notification if ERM_CR1[ENCIE13] is enabled.</p>                                                                                                                                                                                                                           |

*Table continues on the next page...*

## ERM\_SR1 field descriptions (continued)

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 13 mapping.</p> <p>0 No non-correctable error event on Memory 13 detected</p> <p>1 Non-correctable error event on Memory 13 detected</p>                                                                                                                                                                                                                                      |
| 22–23<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                             |
| 24<br>SBC14       | <p>Memory 14 Single-Bit Correction Event</p> <p>This field is initialized by hardware reset.</p> <p>Write 1 to clear this field. This write also clears the corresponding interrupt notification if ERM_CR1[ESCIE14] is enabled.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 14 mapping.</p> <p>0 No single-bit correction event on Memory 14 detected</p> <p>1 Single-bit correction event on Memory 14 detected</p> |
| 25<br>NCE14       | <p>Memory 14 Non-Correctable Error Event</p> <p>This field is initialized by hardware reset.</p> <p>Write 1 to clear this field. This write also clears the corresponding interrupt notification if ERM_CR1[ENCIE14] is enabled.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 14 mapping.</p> <p>0 No non-correctable error event on Memory 14 detected</p> <p>1 Non-correctable error event on Memory 14 detected</p> |
| 26–27<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                             |
| 28<br>SBC15       | <p>Memory 15 Single-Bit Correction Event</p> <p>This field is initialized by hardware reset.</p> <p>Write 1 to clear this field. This write also clears the corresponding interrupt notification if ERM_CR1[ESCIE15] is enabled.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 15 mapping.</p> <p>0 No single-bit correction event on Memory 15 detected</p> <p>1 Single-bit correction event on Memory 15 detected</p> |
| 29<br>NCE15       | <p>Memory 15 Non-Correctable Error Event</p> <p>This field is initialized by hardware reset.</p> <p>Write 1 to clear this field. This write also clears the corresponding interrupt notification if ERM_CR1[ENCIE15] is enabled.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 15 mapping.</p> <p>0 No non-correctable error event on Memory 15 detected</p> <p>1 Non-correctable error event on Memory 15 detected</p> |
| 30–31<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                             |

### 18.3.6 ERM Status Register (ERM\_SR2)

The ERM Status Register is a 32-bit control register for signaling which types of ECC events have been detected for each memory channel. The register signals the last memory event to be detected for memory channels 16 to 19.

Address: 0h base + 18h offset = 18h

|       |       |       |    |    |       |       |    |    |       |       |    |    |       |       |    |    |
|-------|-------|-------|----|----|-------|-------|----|----|-------|-------|----|----|-------|-------|----|----|
| Bit   | 0     | 1     | 2  | 3  | 4     | 5     | 6  | 7  | 8     | 9     | 10 | 11 | 12    | 13    | 14 | 15 |
| R     | SBC16 | NCE16 | 0  |    | SBC17 | NCE17 | 0  |    | SBC18 | NCE18 | 0  |    | SBC19 | NCE19 | 0  |    |
| W     | w1c   | w1c   |    |    | w1c   | w1c   |    |    | w1c   | w1c   |    |    | w1c   | w1c   |    |    |
| Reset | 0     | 0     | 0  | 0  | 0     | 0     | 0  | 0  | 0     | 0     | 0  | 0  | 0     | 0     | 0  | 0  |
| Bit   | 16    | 17    | 18 | 19 | 20    | 21    | 22 | 23 | 24    | 25    | 26 | 27 | 28    | 29    | 30 | 31 |
| R     | 0     |       |    |    |       |       |    |    |       |       |    |    |       |       |    |    |
| W     |       |       |    |    |       |       |    |    |       |       |    |    |       |       |    |    |
| Reset | 0     | 0     | 0  | 0  | 0     | 0     | 0  | 0  | 0     | 0     | 0  | 0  | 0     | 0     | 0  | 0  |

**ERM\_SR2 field descriptions**

| Field      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>SBC16 | <p>Memory 16 Single-Bit Correction Event</p> <p>This field is initialized by hardware reset.</p> <p>Write 1 to clear this field. This write also clears the corresponding interrupt notification if ERM_CR2[ESCIE16] is enabled.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 16 mapping.</p> <p>0 No single-bit correction event on Memory 16 detected<br/>1 Single-bit correction event on Memory 16 detected</p> |
| 1<br>NCE16 | <p>Memory 16 Non-Correctable Error Event</p> <p>This field is initialized by hardware reset.</p> <p>Write 1 to clear this field. This write also clears the corresponding interrupt notification if ERM_CR2[ENCIE16] is enabled.</p> <p><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 16 mapping.</p>                                                                                                                       |

*Table continues on the next page...*



## ERM\_SR2 field descriptions (continued)

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | 0 No non-correctable error event on Memory 16 detected<br>1 Non-correctable error event on Memory 16 detected                                                                                                                                                                                                                                                                                                                       |
| 2–3<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                             |
| 4<br>SBC17        | Memory 17 Single-Bit Correction Event<br>This field is initialized by hardware reset.<br>Write 1 to clear this field. This write also clears the corresponding interrupt notification if ERM_CR2[ESCIE17] is enabled.<br><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 17 mapping.<br>0 No single-bit correction event on Memory 17 detected<br>1 Single-bit correction event on Memory 17 detected |
| 5<br>NCE17        | Memory 17 Non-Correctable Error Event<br>This field is initialized by hardware reset.<br>Write 1 to clear this field. This write also clears the corresponding interrupt notification if ERM_CR2[ENCIE17] is enabled.<br><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 17 mapping.<br>0 No non-correctable error event on Memory 17 detected<br>1 Non-correctable error event on Memory 17 detected |
| 6–7<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                             |
| 8<br>SBC18        | Memory 18 Single-Bit Correction Event<br>This field is initialized by hardware reset.<br>Write 1 to clear this field. This write also clears the corresponding interrupt notification if ERM_CR2[ESCIE18] is enabled.<br><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 18 mapping.<br>0 No single-bit correction event on Memory 18 detected<br>1 Single-bit correction event on Memory 18 detected |
| 9<br>NCE18        | Memory 18 Non-Correctable Error Event<br>This field is initialized by hardware reset.<br>Write 1 to clear this field. This write also clears the corresponding interrupt notification if ERM_CR2[ENCIE18] is enabled.<br><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 18 mapping.<br>0 No non-correctable error event on Memory 18 detected<br>1 Non-correctable error event on Memory 18 detected |
| 10–11<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                             |
| 12<br>SBC19       | Memory 19 Single-Bit Correction Event<br>This field is initialized by hardware reset.<br>Write 1 to clear this field. This write also clears the corresponding interrupt notification if ERM_CR2[ESCIE19] is enabled.                                                                                                                                                                                                               |

*Table continues on the next page...*

**ERM\_SR2 field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | <b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 19 mapping.<br>0 No single-bit correction event on Memory 19 detected<br>1 Single-bit correction event on Memory 19 detected                                                                                                                                                                                                                          |
| 13<br>NCE19       | Memory 19 Non-Correctable Error Event<br>This field is initialized by hardware reset.<br>Write 1 to clear this field. This write also clears the corresponding interrupt notification if ERM_CR2[ENCIE19] is enabled.<br><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory 19 mapping.<br>0 No non-correctable error event on Memory 19 detected<br>1 Non-correctable error event on Memory 19 detected |
| 14–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                             |

**18.3.7 ERM Memory  $n$  Error Address Register (ERM\_EAR $n$ )**

The ERM\_EAR $n$  is a 32-bit register for capturing the address of the last ECC event in Memory  $n$ , where  $n$  denotes the memory channel. Any attempted write to ERM\_EAR $n$  is ignored.

Address: 0h base + 100h offset + (16d  $\times$  i), where i=0d to 19d

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15  | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    | EAR |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**ERM\_EAR $n$  field descriptions**

| Field       | Description                                                                                                                                                                                                             |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>EAR | Memory $n$ Error Address - This field contains the faulting system address of the last recorded ECC event on Memory $n$ .<br><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory $n$ mapping. |

### 18.3.8 ERM Memory $n$ Syndrome Register (ERM\_SYN $n$ )

The ERM\_SYN $n$  is a 32-bit register for capturing the calculated syndrome of the last ECC event on Memory  $n$ , where  $n$  denotes the memory channel. Any attempted write to ERM\_SYN $n$  is ignored. The syndrome value identifies the pertinent bit position on a correctable, single-bit data inversion or a non-correctable, single-bit address inversion. The syndrome value does not provide any additional diagnostic information on non-correctable, multi-bit inversions.

Address: 0h base + 104h offset + (16d × i), where i=0d to 19d

|       |          |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0        | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | SYNDROME |   |   |   |   |   |   |   | 0 |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |          |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**ERM\_SYN $n$  field descriptions**

| Field            | Description                                                                                                                                                                                                              |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–7<br>SYNDROME  | Memory $n$ Syndrome - This field contains the ECC syndrome associated with the last recorded ECC event on Memory $n$ .<br><br><b>NOTE:</b> Refer to the chip-specific ERM information for details on Memory $n$ mapping. |
| 8–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                  |

## 18.4 Functional description

### 18.4.1 Single-bit correction events

When a single-bit correction event on Memory  $n$  is detected, the ERM:

- Records the event by changing to 1 the value of the applicable Status Register bit: ERM\_SR $x$ [SBC $n$ ]
- Records the corresponding access address that initiated the event in the Memory  $n$  Error Address Register: ERM\_EAR $n$
- Stores the corresponding ECC syndrome in the Memory  $n$  Error Syndrome Register: ERM\_SYN $n$ . This register identifies the bit position of the corrected data on single-bit data inversion.

The ERM holds event information only for the last reported event.

To clear the record of an event, write 1 to ERM\_SRx[SBCn] to change its value to 0.

#### 18.4.1.1 Optional interrupt notification for single-bit correction events

The ERM provides an option to generate an interrupt notification upon the report of a single-bit correction event. This sequence describes how to use the option:

1. To enable interrupt notifications for single-bit correction events on Memory  $n$ , set ERM\_CRx[ESCIEn] to 1.
2. Subsequently, when a single-bit correction event on Memory  $n$  is detected, the ERM:
  - Records the event and address and stores the ECC syndrome as usual
  - Additionally sends an interrupt notification corresponding to the event
3. To clear both the record of an event and the corresponding interrupt notification, write 1 to ERM\_SRx[SBCn] to change its value to 0.

#### 18.4.2 Non-correctable error events

When a non-correctable ECC error event on Memory  $n$  is detected, the ERM:

- Records the event by changing to 1 the value of the applicable Status Register bit: ERM\_SRx[NCEn]
- Records the corresponding access address that initiated the event in the Memory  $n$  Error Address Register: ERM\_EAR $n$
- Stores the corresponding ECC syndrome in the Memory  $n$  Error Syndrome Register: ERM\_SYN $n$ 
  - In the event of a non-correctable address bit inversion, ERM\_SYN $n$  identifies the pertinent address bit position.
  - In the event of a non-correctable, multi-bit data inversion, the syndrome value does not provide any additional diagnostic information.

The ERM holds event information only for the last reported event.

To clear the record of an event, write 1 to ERM\_SRx[NCEn] to change its value to 0.

### 18.4.2.1 Optional interrupt notification for non-correctable error events

The ERM provides an option to generate an interrupt notification upon the report of a non-correctable ECC event. This sequence describes how to use the option:

1. To enable interrupt notifications for non-correctable error events on Memory  $n$ , set `ERM_CRx[ENCIE $n$ ]` to 1.
2. Subsequently, when a non-correctable error event on Memory  $n$  is detected, the ERM:
  - Records the event and address and stores the ECC syndrome as usual
  - Additionally sends an interrupt notification corresponding to the event
3. To clear both the record of an event and the corresponding interrupt notification, write 1 to `ERM_SRx[NCE $n$ ]` to change its value to 0.

## 18.5 Initialization

For each ERM channel, prepare the corresponding memory array before enabling ERM interrupts about errors for that memory.

1. Initialize the memory to a known value so that the correct corresponding ECC codeword is stored.
2. During the memory's initialization, if the ERM captures information about any ECC error event, clear the corresponding `ERM_SRx[SBC $n$ ]` or `ERM_SRx[NCE $n$ ]` field that stores the record of the event.
3. Program the applicable `ERM_CRx[ESCI $n$ ]` and `ERM_CRx[ENCIE $n$ ]` fields to enable ERM interrupts as desired.



# Chapter 19

## Clocking

### 19.1 Introduction

This chapter describes the architecture for the system level clocks and includes the following information:

- System clock specifics
- Clock architecture
- Clock sources
- Clock monitoring
- Programmable clock dividers
- Clock control registers

The clock architecture of this device contains two clock domains coming from separate PLLs. The first one is the system clock for the cores, cross-bar switch, peripheral bridges, memories, debug logic and memory mapped portion of peripherals. The second one feeds the state machines and protocol engines of communication and timer peripheral modules. These two clock domains can be configured to be completely asynchronous between each other. The PLL generating the system clock (PLL1) can be configured to be frequency modulated to reduce electromagnetic emissions. The PLL generating the peripheral clock (PLL0) cannot be modulated. It is intended to provide an accurate reference clock for peripheral timer and communication functions that cannot tolerate frequency modulation.

The reference clock for PLL0 (non-FM) may come from the following possible sources:

- Crystal oscillator (XOSC)
- External clock (when XOSC is bypassed)
- Internal 16 MHz RC oscillator (IRC)

The reference clock for PLL1 (FM-modulated) may come from the following possible sources:

- Crystal oscillator (XOSC)
- External clock (when XOSC is bypassed)
- Output of PLL0

Out of reset, both clock domains (system and peripheral) are sourced directly from the IRC. After reset, software can select the PLLs or other clock sources (XOSC, external) to connect to system and peripheral clocks. A Progressive Clock Frequency Switch (PCFS) module is available after the system clock selector to allow gradual frequency changes that minimize transient currents. Clock Monitor Units (CMUs) are connected to outputs of clock dividers to verify that their clock frequencies stay within necessary operating limits. When an issue with a clock is found, the CMU sends a failure signal directly to the FCCU.

## **19.2 Clock generation**

The top-level clock generation architecture can be seen in the figure below. All clock selectors and dividers are located and programmed in memory mapped registers of the SIU.



Notes: 1. The clock divider range in the PLL outputs (phi) is non-contiguous in by-2 steps: [2, 4, ..., 124, 126].

2. Divider values and mux selectors are located in SIU registers:  
SIU\_SYSDIV, SIU\_SDCLKCFG, SIU\_LFCLKCFG, SIU\_PSCLKCFG

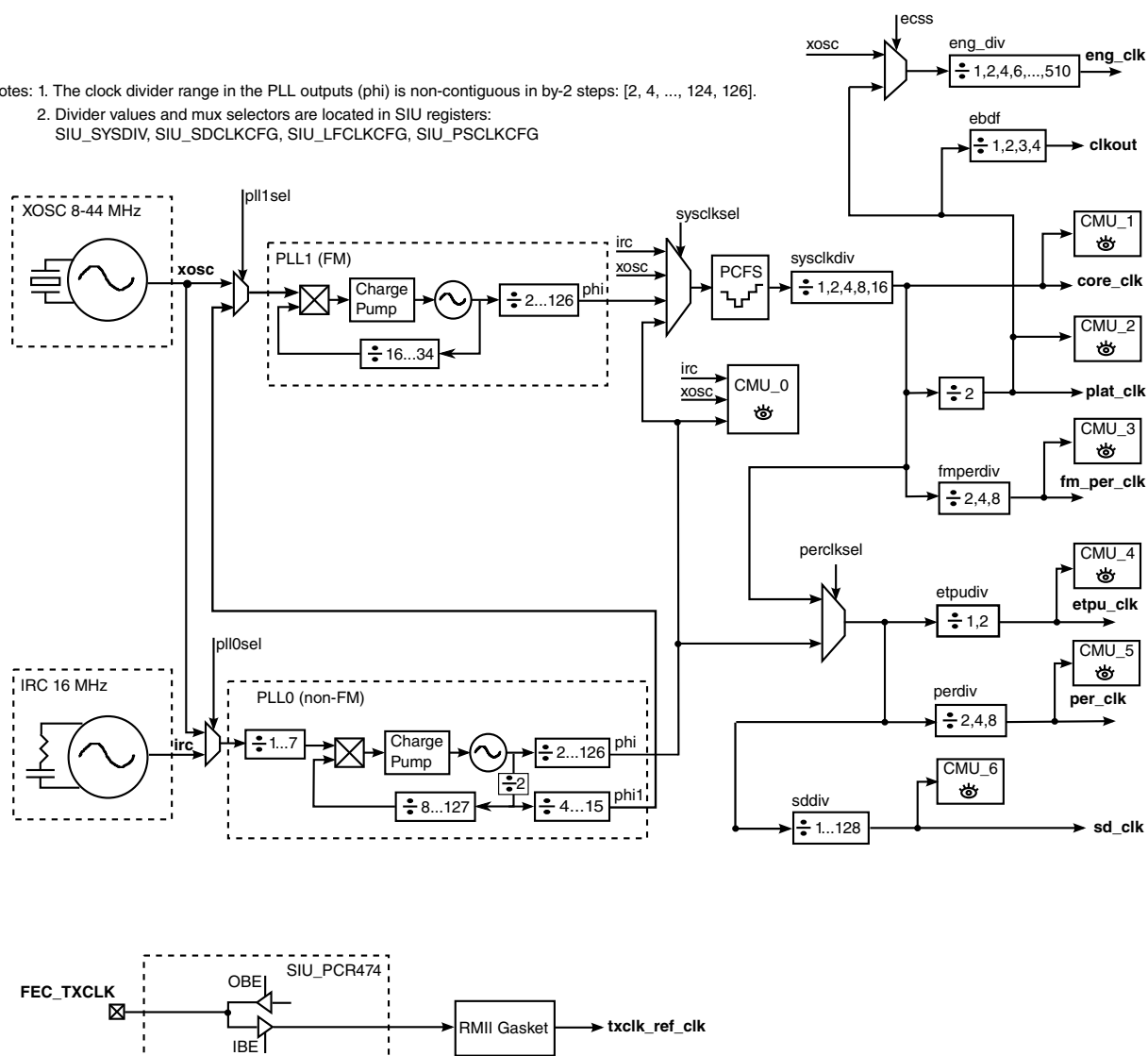


Figure 19-1. Clock generation diagram

## 19.3 SIU registers

The following table shows the relationship between the clocks of the clock generation diagram and the SIU registers.

Table 19-1. Relationship between clocks and SIU registers

| Clock    | Clock destination          | Clock selector       | Clock divider        | CMU                |
|----------|----------------------------|----------------------|----------------------|--------------------|
| PLL0     | PLL0 input                 | SIU_SYSDIV.PLL0SEL   | Internal to PLL0     |                    |
| PLL1     | PLL1 input                 | SIU_SYSDIV.PLL1SEL   | Internal to PLL1     |                    |
| core_clk | Core0, Core1, Checker Core | SIU_SYSDIV.SYSCLKSEL | SIU_SYSDIV.SYSCLKDIV | CMU_1 <sup>1</sup> |

Table continues on the next page...

**Table 19-1. Relationship between clocks and SIU registers (continued)**

| Clock      | Clock destination                                                                       | Clock selector       | Clock divider         | CMU                |
|------------|-----------------------------------------------------------------------------------------|----------------------|-----------------------|--------------------|
| plat_clk   | Platform modules: Cross-Bar, Peripheral Bridges, Memories, DMA, Flash Controller, Debug |                      | Fixed at divided-by-2 | CMU_2 <sup>1</sup> |
| fm_per_clk | Memory mapped registers of peripherals                                                  |                      | SIUSYSDIV.FMPERDIV    | CMU_3              |
| per_clk    | Protocol and communication engines of peripherals                                       | SIU_SYSDIV.PERCLKSEL | SIU_SYSDIV.PERDIV     | CMU_5              |
| etpu_clk   | eTPU A/B/C                                                                              |                      | SIU_SYSDIV.ETPUDIV    | CMU_4              |
| sd_clk     | Sigma-Delta ADC                                                                         |                      | SIU_SDCLKCFG.SDDIV    | CMU_6              |
| MCAN       | MCAN Protocol Engine Clock                                                              | SIU_SYSDIV.MCANSEL   | Internal to MCAN      |                    |
| eng_clk    | Engineering clock pin                                                                   | SIU_ECCR.ECSS        | SIU_ECCR.ENGDIV       |                    |

1. CMU\_1 and CMU\_2 are physically connected to clock tree spots inside the checker core lake.

## 19.4 Operating frequencies

The device supports the maximum operating frequencies given in the following table.

**Table 19-2. Maximum system and peripheral clock frequencies**

| Clock domain                                        | Mode   | Core (CORE_CLK)  | eTPU (eTPU_CLK)  | Platform (PLAT_CLK) | Peripherals memory mapped registers (FM_PER_CLK) | Peripherals protocol engines (PER_CLK) |
|-----------------------------------------------------|--------|------------------|------------------|---------------------|--------------------------------------------------|----------------------------------------|
| Backwards compatible: one clock domain <sup>1</sup> | Double | 264 <sup>2</sup> | 132 <sup>3</sup> | 132 <sup>3</sup>    | 132 <sup>3</sup>                                 | 132 <sup>3</sup>                       |
|                                                     | eTPU   | 200              | 200              | 100                 | 100                                              | 100                                    |
| Enhanced: dual clock domain <sup>4</sup>            | Normal | 264 <sup>2</sup> | 200              | 132 <sup>3</sup>    | 132 <sup>3</sup>                                 | 100                                    |

1. Clock selectors configured such that all clocks come from a single source, either PLL0 or PLL1.

2. For MPC5775B the value is 220 MHz

3. For MPC5775B the value is 110 MHz

4. Clock selectors configured such that both PLLs drive the clocks into two asynchronous domains.

### NOTE

The essentially asynchronous clock domains of EVTO (eTPU\_CLK) and NPC (CORE\_CLK) require CORE\_CLK to be set to a higher frequency than eTPU\_CLK for guaranteed EVTO capture by NPC. Because CORE\_CLK can be set to FM (2% intentional jitter), the frequency of CORE\_CLK minus 2% must be greater than the frequency of eTPU\_CLK.

## 19.5 Default clock configuration

At power up, the IRC is the default clock for the entire system, with clock selectors configured such that all clocks come from a single clock domain. The default configuration after reset for all clock sources, selectors, and dividers is shown in the tables below.

**Table 19-3. Reset configuration of clock sources**

| Source              | Default after reset                                                                                                                           |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| IRC                 | Always enabled; it is the default clock source.                                                                                               |
| XOSC/External clock | Crystal oscillator and external bypass default state depends on DCF record or configuration pins (see SIU_RSR subsection of the SIU chapter). |
| PLL0                | Disabled and powered down                                                                                                                     |
| PLL1                | Disabled and powered down                                                                                                                     |

**Table 19-4. Reset configuration of clock selectors**

| Selector              | Default after reset |
|-----------------------|---------------------|
| SIU_SYSDIV.PLL0SEL    | Selects XOSC        |
| SIU_SYSDIV.PLL1SEL    | Selects XOSC        |
| SIU_SYSDIV.SYSCLKSEL  | Selects the IRC     |
| SIU_SYSDIV.PERCLKSEL  | Selects core_clk    |
| SIU_LFCLKCFG.LFCLKSEL | Selects the XOSC    |
| SIU_SYSDIV.MCANSEL    | Selects the XOSC    |
| SIU_ECCR.ECSS         | Selects plat_clk    |

**Table 19-5. Reset configuration of clock dividers**

| Divider              | Default after reset |
|----------------------|---------------------|
| SIU_SYSDIV.SYSCLKDIV | Divide-by-1         |
| SIUSYSDIV.FMPERDIV   | Divide-by-2         |
| SIU_SYSDIV.PERDIV    | Divide-by-2         |
| SIU_SYSDIV.ETPUDIV   | Divide-by-2         |
| SIU_SDCLKCFG.SDDIV   | Divide-by-1         |
| SIU_PSCLKCFG.PSDIV   | Divide-by-1         |
| SIU_PSCLKCFG.PSDIV1M | Divide-by-1         |
| SIU_LFCLKCFG.LFDIV   | Divide-by-1         |
| SIU_ECCR.EBDF        | Divide-by-2         |
| SIU_ECCR.ENGDIV      | Divide-by-32        |

## 19.6 Clock sources

This chip has these clock sources:

- 16 MHz Internal RC oscillator (IRC)
- External oscillator (XOSC)
- PLL0 (non-FM PLL)
- PLL1 (FMPLL)

### 19.6.1 16 MHz RC oscillator (IRC)

This device has a 16 MHz internal RC oscillator that is always enabled and cannot be powered down. The IRC can be used as the source of system clock, peripheral clock and PLL clock references. The IRC is the default system and peripheral clock after reset.

The IRC has no dedicated software accessible register interface.

### 19.6.2 External oscillator (XOSC)

The external oscillator (XOSC) provides a clock source for the on-chip PLLs and some peripherals that require accurate clock reference (FlexCAN, MCAN, PIT, PLL, CMU\_0). If required, it can also be used directly as a clock source to the main system and peripheral clocks. The XOSC does not have any software accessible register interface. Configuration is done using DCF records read from UTEST flash memory, which are read by the SSCM during reset and driven to the oscillator. Changing the XOSC configuration requires a reset cycle to take effect.

#### NOTE

The DCF configuration bits for the XOSC have triple-voting flip-flop (TVF) implementation to prevent undetectable errors in the system. See the Device Configuration Format chapter for more information on DCF programming.

#### 19.6.2.1 XOSC configuration

The XOSC has the following configurable features:

- Crystal oscillator can be bypassed to source a clock directly from the EXTAL pin (bypass mode)
- Internal load capacitors can be selected for 20/40 MHz crystal oscillators
- Automatic Loop Control (ALC) can be applied to avoid the need of external series resistor
- Oscillator gain can be selected to match the crystal frequency range

Bypass configuration is controlled by bit 24 of the Miscellaneous DCF record. The selection of internal or external load capacitors on the XTAL/EXTAL pins is determined by bit 25 of the Miscellaneous DCF record. Load capacitor values are determined by the crystal manufacturer data sheet requirements, while accounting for stray PCB and on-chip capacitance. See the device data sheet for on-chip capacitance values.

When using internal load capacitors for the oscillator, the internal capacitance value is configured in the range of 1–16 pF, in approximate steps of 1 pF, by bits 26–30 of the Miscellaneous DCF record. The default value for the trimming bits (0b000000) selects the minimum internal capacitance value (about 1.8 pF).

The XOSC has the ability to be started with a crystal of 8 MHz, 20 MHz, or 40 MHz. Bits 18 and 21 of the Miscellaneous DCF record select which crystal source to use.

Automatic Loop Control (ALC) can be enabled by bit 17 of the Miscellaneous DCF record.

### 19.6.2.2 XOSC startup

During reset, the SSCM reads the XOSC\_CNT field from the Miscellaneous DCF record and pre-loads this value to an internal counter clocked by XOSC. Upon reset exit, the counter counts 512 times XOSC\_CNT. When counting finishes, it means that the XOSC clock is considered to be stable, and a flag called XOSC is set in the SIU\_RSR register. This is a “set-once” bit, which software can check before attempting to use the external oscillator clock source soon after a reset cycle. This flag also enables CMU0 to start monitoring the XOSC clock. From this moment on, if the XTAL clock fails, the SIU\_RSR[XOSC] flag will not be negated to indicate the fail condition, but CMU0 will set a flag and also send an event signal to the FCCU.

### 19.6.3 PLL

This chip has a dual PLL subsystem that provides separate system and peripheral clocks.

The PLLs are disabled after reset. Enable them by software before selecting them as clock sources.

The following image is a block diagram of the dual PLL subsystem.

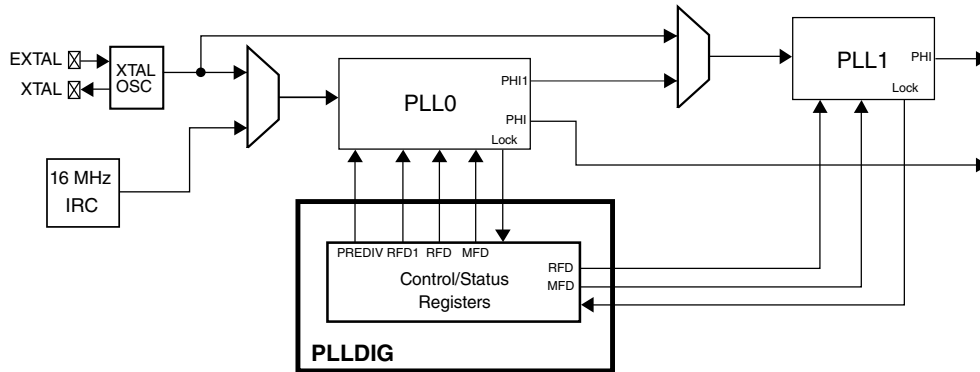


**Figure 19-2. Dual PLL digital interface block diagram**

### 19.6.3.1 PLL0 (non-FM)

PLL0 is the primary PLL. It provides a non-Frequency Modulated clock to the peripherals and the reference clock to PLL1.

#### 19.6.3.1.1 PLL0 input clocks

The possible input clock sources for PLL0 are the XOSC and IRC. Select the clock reference using multiplexers controlled by the SIU\_SYSDIV register.

#### 19.6.3.1.2 PLL0 output clocks

The output clocks from PLL0 are PHI and PHI1.

- The PHI output clock drives various peripheral clocks and the system clock when PLL1 is not locked or active.
- The PHI1 output provides one of the input references for PLL1.

To generate the reference clock to PLL1, PLL0 has a dedicated output clock divider that is separate from the divider that generates the clock to peripherals.

### 19.6.3.2 PLL1 (FM)

PLL1 is a Frequency Modulated PLL (FMPLL) that is used to drive the system clock.

### 19.6.3.2.1 PLL1 input clocks

The possible input clock sources for PLL1 are PLL0 and the XOSC. Select the clock reference using multiplexers controlled by the SIU\_SYSDIV register.

### 19.6.3.2.2 PLL1 output clocks

The output clock from PLL1 drives the system clock. The PHI output clock contains a fractional divider that can be applied to the loop divide of the PLL to achieve good granularity in the PLL1 PHI output clock frequency.

### 19.6.3.3 PLL register interface

This chip has a single digital interface for both PLLs. The digital interface provides register control of all available features of PLL0 and PLL1.

Before changing any of the dividers or FM configuration, you must force the PLLs into a disabled state by writing 00b to the CLKCFG field of the PLL0CR and PLL1CR registers. This method is the only way to power down the PLLs.

### 19.6.3.4 PLLDIG initialization information

Immediately after reset, PLL0 and PLL1 are disabled. The recommended procedure to program the PLLs and enter normal mode is:

1. Configure PLL0 and related modules.
  - a. With PLL0 disabled, program the desired PLL0 clock source in SIU\_SYSDIV[PLL0SEL]. The default source is XOSC. Write 1 to SIU\_SYSDIV[PLL0SEL] to select IRC as the source.
  - b. Program PLL0DV[PREDIV], PLL0DV[RFDPHI1], PLL0DV[MFD], and PLL0DV[RFDPHI].
  - c. If necessary, modify the CMU frequency meter values (CMU\_MDR and CMU\_FDR) used to monitor the XOSC frequency. The XOSC frequency is compared to the IRCOSC source.
2. Ensure SIU\_RSR[XOSC] is 1 to confirm that XOSC is stable.
3. Turn on PLL0.
  - a. Enable PLL0 by setting PLL0CR[CLKCFG] to 11b.

- b. Poll PLL0SR[LOCK] until its value is 1.
4. Configure PLL1 and related modules.
  - a. With PLL1 disabled, program the PLL1 clock source in SIU\_SYSDIV[PLL1SEL]. The default source is XOSC. Write 1 to SIU\_SYSDIV[PLL1SEL] to select the PHI1 output of PLL0 as the source.
  - b. Program PLL1DV[MFD] and PLL1DV[RFDPHI].
  - c. If required:
    - Program the PLL1FM register with the desired frequency modulation parameters, and
    - Enable FM modulation by setting PLL1FM[MODEN] to 1.

The PLL1FM register must not be written after PLL1 is enabled and operating in normal mode.
5. Turn on PLL1.
  - a. Enable PLL1 by setting PLL1CR[CLKCFG] to 11b.
  - b. Poll PLL1SR[LOCK] until its value is 1.
6. Source the system clock and peripheral clock from the PLLs by programming SIU\_SYSDIV[SYSCLOCKSEL] and SIU\_SYSDIV[PERCLKSEL], respectively.

## 19.7 Clock gating

The SIU\_HLT1 and SIU\_HLT2 registers provide software-controlled clock gating. These registers dispatch request signals to various modules to halt their clock inputs. Modules respond differently to the halt request. The CPU allows its clock inputs to be halted only after the execution of the WAIT instruction. Most other modules wait for internal activities to finish before allowing their clock inputs to be halted by asserting a halt acknowledge signal. A few modules do not implement the stop acknowledge protocol within them and allow their clocks to be halted immediately after the halt request is asserted.

Some peripherals have an internal memory mapped register that can be used to disable the module and to halt the clock input to the module. In most cases the peripheral registers of disabled modules are still readable and writable. However, the halt mechanism of the SIU\_HALT registers disables the read/write functions of the peripheral registers for additional power saving.



When the clocks of all CPUs have been halted and all XBAR masters are idle, then the clocks of the following modules are also halted automatically: XBAR, Peripheral Bridges, system RAM, DMA, STM and Semaphores. The INTC and SWT\_A/SWT\_B clocks are never stopped.

## 19.8 Core and on-platform clocking

The clocks to the cores and on-platform modules come from the FM PLL. The core clock (CORE\_CLK) is double the frequency of the platform clock (PLAT\_CLK), but they are synchronous to each other (same clock domain). The frequency ratio between CORE\_CLK and PLAT\_CLK is fixed at 2:1.

The following modules belong to the core clock domain:

- Core0
- Core1
- Checker core
- Cache memories
- Nexus Port Controller (NPC)

The following modules belong to the platform clock domain:

- System RAM and RAM Controller
- Flash memory and Flash Memory Controller
- Crossbar Switch
- DMA

The frequency modulated peripheral clock (FM\_PER\_CLK) mediates the communication between the Peripheral Bridges and memory-mapped registers within peripheral modules. This clock is part of the same FM clock domain as the core and platform clocks.

## 19.9 Peripheral clocks

Some modules have two clock domains implemented within them. One domain is for the state machines (protocol and/or timing engines) that require accurate non-FM clock sources, and the other domain is for the memory mapped registers that must be accessed by the Peripheral Bridges at the FM clock domain. These modules with separate clock domains can be connected directly to the two FM and non-FM clock sources.

Some other modules, like the eTPU, have a single clock domain with a single clock input.

The following figure depicts all of these cases.

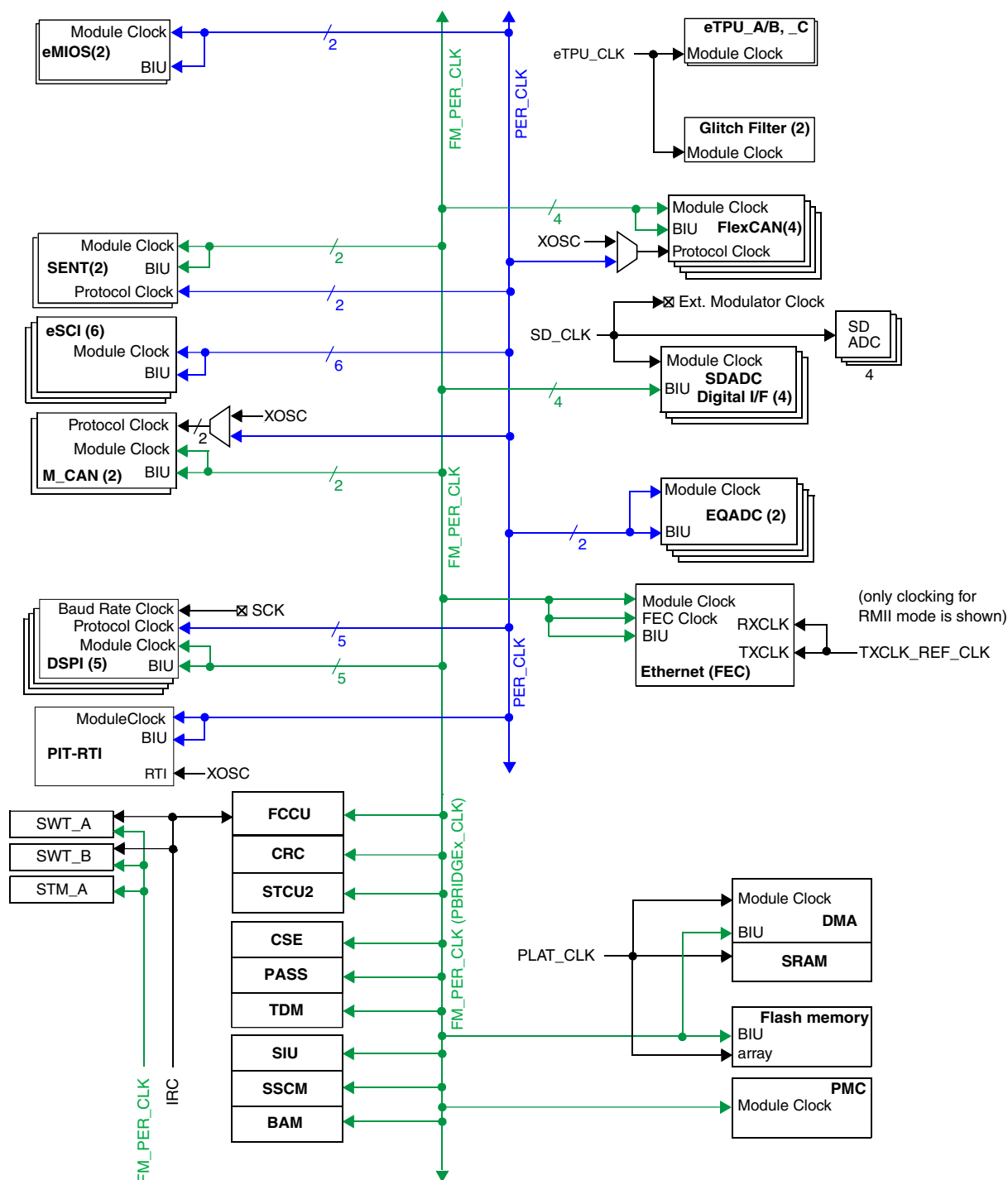


Figure 19-3. Clock distribution

### 19.9.1 eTPU clocking

The eTPUs have a separate clock domain specific to them (ETPU\_CLK), coming from the non-FM PLL. The eTPU clock is synchronous to the other non-FM peripheral clock domain (PER\_CLK), but is asynchronous to the FM-domain clocks (CORE\_CLK, PLAT\_CLK, FM\_PER\_CLK). The reason for a separate clock domain to the eTPUs is to allow them to run at higher speeds than other peripherals, thus optimizing performance and power.

The eTPU module itself has just one clock input that is used by all internal circuits, including timing engines and memory mapped registers. Because this single clock input is connected to the non-FM clock domain, a synchronization gasket mediates communications between the eTPUs and the peripheral bridges, which run at the FM-modulated clock domain. The operation of this gasket is transparent to software.

### 19.9.2 FlexCAN and MCAN clocking

The clock source for the CAN protocol engines can optionally come from the peripheral clock or from the XOSC. For the FlexCAN module, selection of the engine clock source is based on a register field internal to the FlexCAN module. Each FlexCAN instance can select a different source.

For the MCAN modules, selection of the engine clock source is based on the SIU\_SYSDIV[MCANSEL] field. In this case, all MCAN instances receive the same clock.

### 19.9.3 Ethernet clocking

Only an external clock for RMII mode is supported. This chip does not generate an FEC clock internally. The external clock comes from a package pin and goes through the RMII gasket.

#### NOTE

While the FEC is operating, do not change clock configurations (muxes or dividers). Otherwise, FEC transactions in the crossbar might be corrupted.

### 19.9.4 SDADC clocking

The SDADC does not support the use of an external modulator. As a result, the only clock available is the internal clock (`sd_clk`), which is derived from the peripheral (non-FM) clock domain through a clock divider. The division factor for this divider is specified by the `SIU_SDCLKCFG[SDDIV]` field.

### 19.10 Serial boot clocking

Because the IRC is not accurate enough for the serial protocols, prior to entering serial boot mode the BAM switches the clock source to the external oscillator, using either the crystal or external reference—whichever one is effective at startup. No direct signal is available to an external host to indicate that serial boot has been initiated by the BAM.

### 19.11 MCKO

MCKO is an output clock from the Nexus Port Controller (NPC) to the development tools used for the timing of MSEO and MDO pin functions. MCKO is derived from the core clock (`core_clk`), and its frequency is determined by the value of the `MCKO_DIV` field in the port configuration register (PCR) located in the NPC.

### 19.12 CLKOUT / ENGCLK

CLKOUT is a device clock output derived from the platform clock and intended for, but not restricted to, use with the External Bus Interface (EBI). ENGCLK is a 50% duty cycle output clock derived from the platform clock or the crystal oscillator. Both clocks belong to the FM-PLL clock domain, but ENGCLK is not synchronous to CLKOUT (not edge aligned).

The external clock pins CLKOUT and ENGCLK are enabled during and immediately after reset. CLKOUT defaults to the platform clock frequency divided by two, and ENGCLK defaults to the platform clock frequency divided by 32.

### 19.13 Clock monitoring

For all safety-relevant clocks, this chip uses [Clock Monitor Unit \(CMU\)](#) modules to detect a missing clock or incorrect frequency. The IRC and XOSC are used as the clock monitor references.

The chip has nine instances of the CMU module. Each CMU module is programmed independently.

#### 19.13.1 CMU-monitored clocks

The following table shows the clock or clocks monitored by each CMU module instance.

**Table 19-6. CMU-monitored clocks**

| Monitor | Monitored clock(s)                                                                            |
|---------|-----------------------------------------------------------------------------------------------|
| CMU_0   | PLL0, XOSC, IRC                                                                               |
| CMU_1   | Core0, Core1, Checker Core, RCCU                                                              |
| CMU_2   | Platform modules: Crossbar, Peripheral Bridges, Memories, DMA, Flash Memory Controller, Debug |
| CMU_3   | Memory-mapped registers of peripherals                                                        |
| CMU_4   | eTPU_A, eTPU_B, eTPU_C                                                                        |
| CMU_5   | Protocol and communication engines of peripherals                                             |
| CMU_6   | SDADC                                                                                         |

#### 19.13.2 CMU\_0 configuration

CMU\_0 automatically monitors the XOSC clock and is able to monitor PLL0.

The following figure is a block diagram for CMU\_0 on this chip.

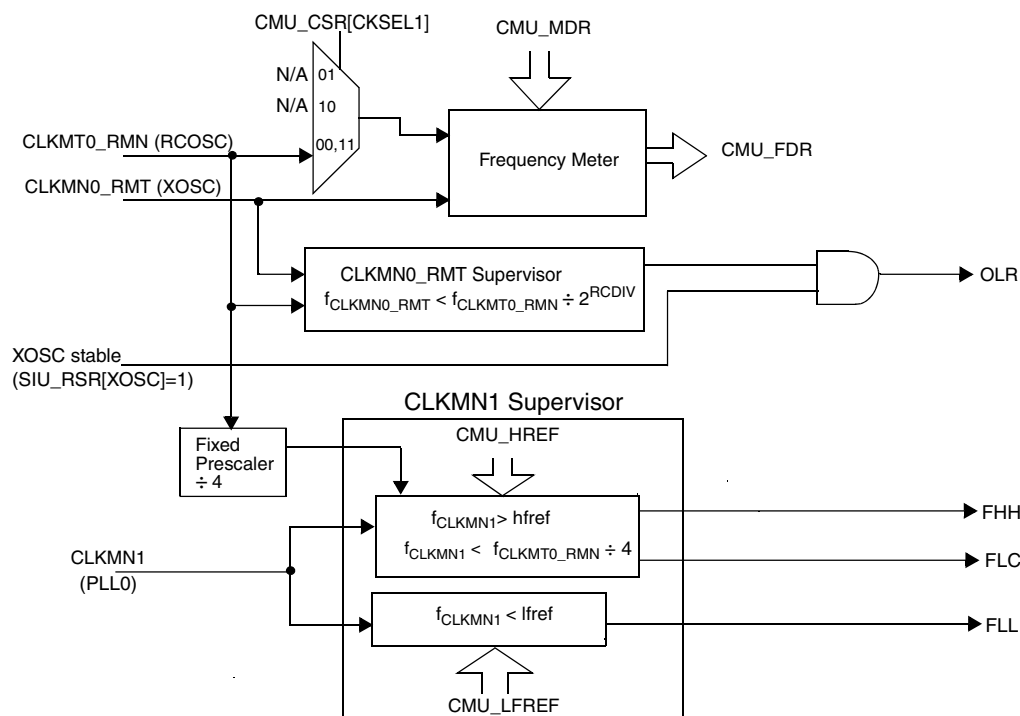


Figure 19-4. CMU\_0 block diagram

### 19.13.2.1 CMU\_0 monitoring of XOSC

CMU\_0 is the only CMU instance that monitors the XOSC clock. CMU\_0 uses the IRC clock to measure whether the XOSC clock's frequency is too slow.

CMU\_0's clock monitoring of the XOSC is not controlled by software. Its monitoring of XOSC is always enabled automatically after the SIU\_RSR[XOSC] field indicates that the XOSC clock is stable.

### 19.13.2.2 CMU\_0 monitoring of PLL0

To enable CMU\_0's monitoring of PLL0, the proper initialization sequence is:

1. Configure and enable PLL0.
2. Wait for PLL0 to lock.
3. Enable CMU\_0 for monitoring by setting the CMU\_0\_CSR[CME] field.

### 19.13.3 CMU\_1 to CMU\_8 configuration

For CMU\_1 to CMU\_8:

- Each CMU operates independently from the module that it monitors, with no hardware interlock between them.
- The IRC is the reference clock.

The following figure is a block diagram for CMU\_1 to CMU\_5 on this chip.

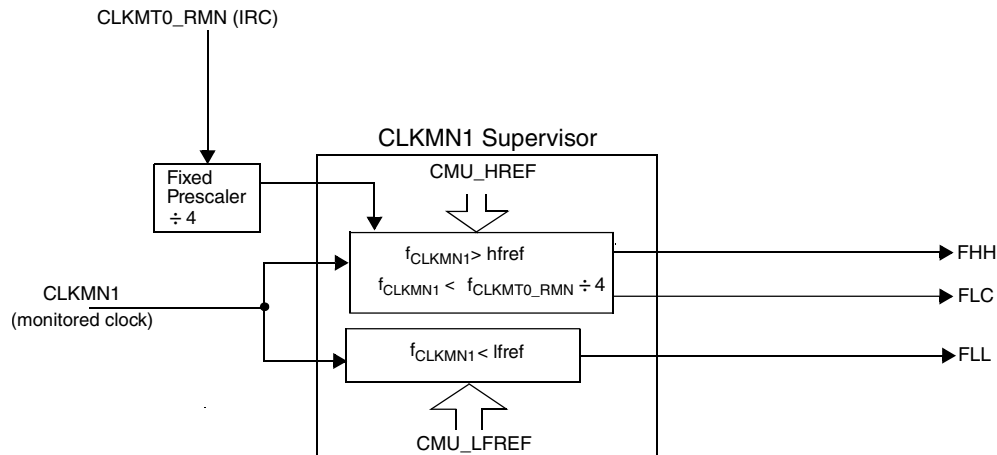


**Figure 19-5. CMU\_1 to CMU\_5 block diagram**

The following figure is a block diagram for CMU\_6 to CMU\_8 on this chip.

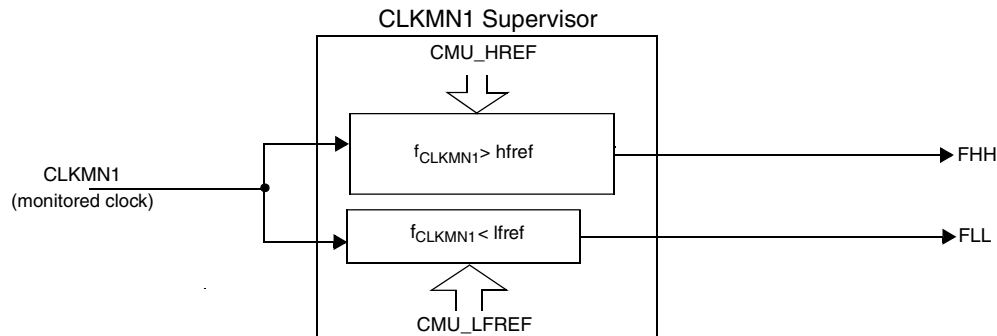


**Figure 19-6. CMU\_6 to CMU\_8 block diagram**

To enable monitoring of each of CMU\_1 to CMU\_8, the proper initialization sequence is:

1. Enable the clock source corresponding to the particular CMU instance.
2. Ensure that the particular clock source is enabled, stable, and ready to be monitored.
  - If the XOSC is the clock source, then the SIU\_RSR[XOSC] flag must be set.
  - If one of the PLLs is the clock source, then that PLL must be locked.



3. Enable the particular CMU instance for monitoring by setting the applicable CMU<sub>x</sub>\_CSR[CME] field.

### 19.13.4 CMU output signals to FCCU and INTC

The output signals of the CMUs are connected to the FCCU in the following way:

- The OLR output of CMU\_0 is connected to the CMU\_0\_OSC channel of the FCCU.
- The FLC, FHH, and FLL outputs of CMU\_0 are combined (ORed) together and then connected to the CMU\_0\_PLL channel of the FCCU.
- The FLC, FHH, and FLL outputs of CMU\_1 to CMU\_3 are all combined (ORed) together and then connected to the CMU\_COMP channel of the FCCU.
- The FLC, FHH, and FLL outputs of CMU\_4 and the FHH and FLL outputs of CMU\_5 to CMU\_8 are all combined (ORed) together and then connected to the CMU\_OTHER channel of the FCCU.

The output signals of the CMUs are also connected to the INTC in the following way:

- All outputs from CMU\_0 and CMU\_1 are combined (ORed) together and then connected to interrupt vector 350.
- All outputs from CMU\_2 and CMU\_3 are combined (ORed) together and then connected to interrupt vector 351.
- All outputs from CMU\_4 and CMU\_5 are combined (ORed) together and then connected to interrupt vector 352.
- All outputs from CMU\_6 and CMU\_7 are combined (ORed) together and then connected to interrupt vector 353.
- All outputs from CMU\_8 are combined (ORed) together and then connected to interrupt vector 354.

### 19.13.5 CMU register and field availability

[Table 19-7](#) specifies which registers and fields are available on a given CMU for this device.

**Table 19-7. CMU register and field availability**

| Address offset | Register   | CMU   | Note                                                                                                                                                                                            |
|----------------|------------|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x0000         | CMU_CSR    | All   | The SFM, CKSEL1, and RCDIV fields are in CMU_0_CSR only.                                                                                                                                        |
| 0x0004         | CMU_FDR    | CMU_0 | —                                                                                                                                                                                               |
| 0x0008         | CMU_HFREFR | All   | —                                                                                                                                                                                               |
| 0x000C         | CMU_LFREFR | All   | —                                                                                                                                                                                               |
| 0x0010         | CMU_ISR    | All   | The OLRI field is in CMU_0_ISR only. The FLCI field is available in ISR for CMU_0 to CMU_5 and not available in ISR CMU_6 to CMU_8. The FHHI and FLLI fields are available in ISR for all CMUs. |
| 0x0014         | Reserved   | —     | —                                                                                                                                                                                               |
| 0x0018         | CMU_MDR    | CMU_0 | —                                                                                                                                                                                               |

### 19.13.6 CMU\_0\_CSR reset value

The chip-specific reset value of CMU\_0\_CSR is 0000\_0000h.

The reset value of CSR for all other CMU instances, and the reset value of other registers for all CMU instances, appear in the CMU's [Register description and memory map](#).

### 19.13.7 CMU use restrictions

The following restrictions apply to CMU use:

- In CMU\_0,  $f_{\text{CLKMN0\_RMT}}$  must be greater than  $f_{\text{CLKMT0\_RMN}} / 2^{\text{RCDIV}}$  by at least 0.5 MHz to guarantee correct CLKMN0\_RMT monitoring. In CMU\_1 to CMU\_5,  $f_{\text{CLKMN1}}$  must be greater than  $f_{\text{CLKMT0\_RMN}} / 4$  by at least 0.5 MHz to guarantee correct CLKMN1 monitoring. The accuracy of CLKMT0\_RMN (IRC) is specified in the chip's Data Sheet.
- If the checker core and RCCU are disabled (via a DCF client), then the clock to the checker core lake is halted to save power. Because CMU\_1 and CMU\_2 are monitoring points in the clock tree inside the checker core lake, these CMUs cannot be used if the checker core is disabled.

### 19.13.8 Loss-of-clock behavior

Loss-of-clock detection is provided by the FLC output signals of the CMUs; see [Figure 19-4](#) and [Figure 19-5](#). The FLC output signals are combined with other CMU output signals and then routed to the FCCU, as explained in [CMU output signals to FCCU and INTC](#). The FCCU can be programmed to react to CMU failures with a system reset.

Because the FCCU is clocked by the IRC, it can generate the reset request even in the absence of the system clock. The reset signal generated by the FCCU overwrites the system clock selection from SIU\_SYSDIV[SYSCLKSEL], forcing the selection of the IRC as the clock source. This mechanism guarantees that the reset state machine continues to receive a (running) clock so that it can propagate the reset request generated by the FCCU.

### NOTE

CMU\_6 to CMU\_8 monitor clocks whose frequency can be lower than the IRC frequency divided by 4, so they do not implement the FLC detection. The other signals (FLL, FHH) do not assert when the monitored clock stops. Therefore these CMUs cannot detect a loss-of-clock condition.

## 19.14 PLL loss of lock interrupt

Always write 1 to both PLL0CR[LOLIE] and PLL1CR[LOLIE] to keep the loss of lock interrupt function enabled.

The main interrupt functionality is determined by the FCCU via fault input NCF[23] for PLL0 and fault input NCF[24] for PLL1.

## 19.15 Debug mode

The clock generation modules (PLL0, PLL1, XOSC, and IRC) and clock monitoring units (CMUs) do not support entry into debug mode. Therefore, there is no possibility of freezing these modules in debug mode.



## Chapter 20

# Progressive Clock Switch (PCS)

### 20.1 Introduction

This chip contains a Progressive Clock Switch (PCS) module that can be used to prevent sudden voltage drops and overshoots when the system clock frequency is switched from one clock source to another.

### 20.2 Block diagram

The following figure is a block diagram of the PCS system.

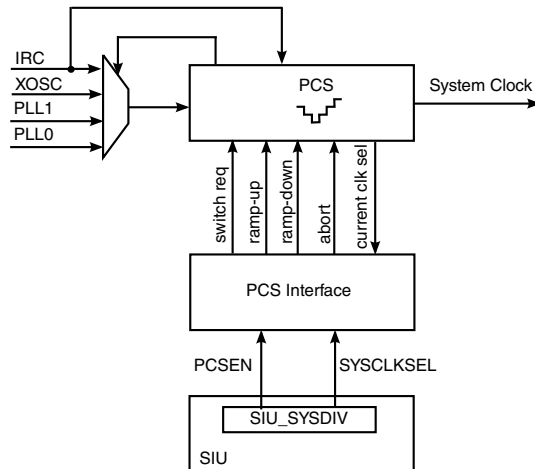


**Figure 20-1. PCS system**

### 20.3 Features

The PCS module provides the following features:

- Progressive frequency switching to minimize power supply voltage drops and overshoots
- Clock switching done in two steps: ramp-down from source frequency to IRC frequency, then ramp-up to target frequency
- Fully programmable ramp-down and ramp-up characteristics
- Automatic monitoring of changes on clock switch selector to trigger progressive clock switching feature

## 20.4 Memory map and register definition

Before programming the PCS registers (PCS\_SDUR, PCS\_DIVC $n$ , PCS\_DIVE $n$ , and PCS\_DIVS $n$ ), select IRC as the system clock: ensure SIU\_SYSDIV[SYSCLKSEL] is 00b. This selection preserves reliable progressive clock operation if, during the PCS register programming, a reset event occurs.

Any access to unused registers and write accesses to read-only registers:

- do not change register content
- cause a transfer error

Unless otherwise noted, all registers may be accessed as 32-bit words, 16-bit half-words, or 8-bit bytes. The byte order is big-endian. For example, the PCS\_DIVC1 register's RATE field may be accessed as a word at address offset 704h, as a half-word at address offset 706h, or as a byte at address offset 707h.

**PCS memory map**

| Address offset (hex) | Register name                                    | Width (in bits) | Access | Reset value | Section/ page               |
|----------------------|--------------------------------------------------|-----------------|--------|-------------|-----------------------------|
| 700                  | PCS Switch Duration Register (PCS_SDUR)          | 32              | R/W    | 0000_0000h  | <a href="#">20.4.1/779</a>  |
| 704                  | PCS Divider Change Register for XOSC (PCS_DIVC1) | 32              | R/W    | 03E7_0000h  | <a href="#">20.4.2/779</a>  |
| 708                  | PCS Divider End Register for XOSC (PCS_DIVE1)    | 32              | R/W    | 0000_03E7h  | <a href="#">20.4.3/780</a>  |
| 70C                  | PCS Divider Start Register for XOSC (PCS_DIVS1)  | 32              | R/W    | 0000_03E7h  | <a href="#">20.4.4/780</a>  |
| 710                  | PCS Divider Change Register for PLL1 (PCS_DIVC2) | 32              | R/W    | 03E7_0000h  | <a href="#">20.4.5/781</a>  |
| 714                  | PCS Divider End Register for PLL1 (PCS_DIVE2)    | 32              | R/W    | 0000_03E7h  | <a href="#">20.4.6/782</a>  |
| 718                  | PCS Divider Start Register for PLL1 (PCS_DIVS2)  | 32              | R/W    | 0000_03E7h  | <a href="#">20.4.7/782</a>  |
| 71C                  | PCS Divider Change Register for PLL0 (PCS_DIVC3) | 32              | R/W    | 03E7_0000h  | <a href="#">20.4.8/783</a>  |
| 720                  | PCS Divider End Register for PLL0 (PCS_DIVE3)    | 32              | R/W    | 0000_03E7h  | <a href="#">20.4.9/783</a>  |
| 724                  | PCS Divider Start Register for PLL0 (PCS_DIVS3)  | 32              | R/W    | 0000_03E7h  | <a href="#">20.4.10/784</a> |

### 20.4.1 PCS Switch Duration Register (PCS\_SDUR)

This register contains the progressive system clock switching duration for each step.

Address: 0h base + 700h offset = 700h

|       |      |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | SDUR |   |   |   |   |   |   |   | 0 |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |      |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0    | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

#### PCS\_SDUR field descriptions

| Field            | Description                                                                                             |
|------------------|---------------------------------------------------------------------------------------------------------|
| 0–7<br>SDUR      | Switch Duration<br>This value defines the duration of one PCS clock switch step in terms of IRC cycles. |
| 8–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                 |

### 20.4.2 PCS Divider Change Register for XOSC (PCS\_DIVC1)

This register defines the rate of frequency change and initial change value for the progressive system clock switching when switching the system clock source to or from the crystal oscillator XOSC on ramp-up and ramp-down, respectively.

#### NOTE

Byte write accesses are not allowed for the INIT field of this register. Such an access will not result in an exception, but the value will not be loaded with the new value.

Address: 0h base + 704h offset = 704h

|       |      |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |    |    |    |
|-------|------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----|----|----|----|----|----|----|----|
| Bit   | 0    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23   | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | INIT |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | 0  |    |    |    |    |    |    | RATE |    |    |    |    |    |    |    |    |
| W     |      |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |    |    |    |
| Reset | 0    | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1  | 0  | 0  | 1  | 1  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

#### PCS\_DIVC1 field descriptions

| Field             | Description                                                                                                                               |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>INIT      | Divider Change Initial Value<br>This is initial change value of the clock divider for the clock ramp-up phase when switching to the XOSC. |
| 16–23<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                   |

Table continues on the next page...

**PCS\_DIVC1 field descriptions (continued)**

| Field         | Description                                                                                                                                                     |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 24–31<br>RATE | Divider Change Rate<br><br>This value controls the change value of the clock divider for the clock ramp-up and ramp-down phase when switching to/from the XOSC. |

**20.4.3 PCS Divider End Register for XOSC (PCS\_DIVE1)**

This register defines the final division value for the progressive system clock switching when switching the system clock source from the XOSC on ramp-down.

**NOTE**

Byte and half-word write accesses are not allowed for this register. Such an access will not result in an exception, but the value will not be loaded with the new value.

Address: 0h base + 708h offset = 708h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 1  | 1  | 1  | 1  | 1  | 0  | 0  | 1  | 1  | 1  |

**PCS\_DIVE1 field descriptions**

| Field            | Description                                                                                                              |
|------------------|--------------------------------------------------------------------------------------------------------------------------|
| 0–11<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                  |
| 12–31<br>DIVE    | Divider End Value<br><br>This is the clock divider end value for the clock ramp-down phase when switching from the XOSC. |

**20.4.4 PCS Divider Start Register for XOSC (PCS\_DIVS1)**

This register defines the initial division value for the progressive system clock switching when switching the system clock source to the XOSC on ramp-up.

**NOTE**

Byte and half-word write accesses are not allowed for this register. Such an access will not result in an exception, but the value will not be loaded with the new value.



Address: 0h base + 70Ch offset = 70Ch

|       |   |   |   |   |   |   |   |   |   |   |    |    |      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12   | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    | DIVS |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 1  | 1  | 1  | 1  | 0  | 0  | 1  | 1  | 1  |

**PCS\_DIVS1 field descriptions**

| Field            | Description                                                                                                                 |
|------------------|-----------------------------------------------------------------------------------------------------------------------------|
| 0–11<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                     |
| 12–31<br>DIVS    | Divider Start Value<br>This is the start value of the clock divider for the clock ramp-up phase when switching to the XOSC. |

**20.4.5 PCS Divider Change Register for PLL1 (PCS\_DIVC2)**

This register defines the rate of frequency change and initial change value for the progressive system clock switching when switching the system clock source to or from PLL1 on ramp-up and ramp-down, respectively.

**NOTE**

Byte write accesses are not allowed for the INIT field of this register. Such an access will not result in an exception, but the value will not be loaded with the new value.

Address: 0h base + 710h offset = 710h

|       |      |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |    |    |    |
|-------|------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----|----|----|----|----|----|----|----|
| Bit   | 0    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23   | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | INIT |   |   |   |   |   |   |   |   |   |    |    |    |    |    | 0  |    |    |    |    |    |    |    | RATE |    |    |    |    |    |    |    |    |
| W     |      |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |    |    |    |
| Reset | 0    | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1  | 0  | 0  | 1  | 1  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**PCS\_DIVC2 field descriptions**

| Field             | Description                                                                                                                                             |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>INIT      | Divider Change Initial Value<br>This is initial change value of the clock divider for the clock ramp-up phase when switching to PLL1.                   |
| 16–23<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                 |
| 24–31<br>RATE     | Divider Change Rate<br>This value controls the change value of the clock divider for the clock ramp-up and ramp-down phase when switching to/from PLL1. |

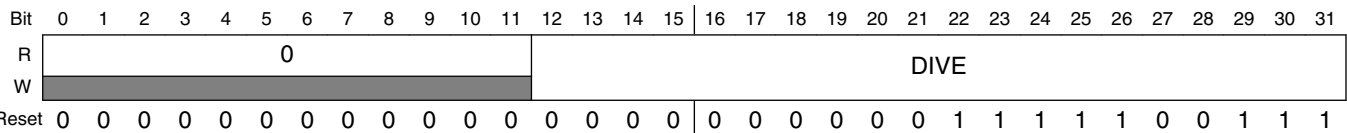
20.4.6 PCS Divider End Register for PLL1 (PCS\_DIVE2)

This register defines the final division value for the progressive system clock switching when switching the system clock source from PLL1 on ramp-down.

NOTE

Byte and half-word write accesses are not allowed for this register. Such an access will not result in an exception, however the value will not be loaded with the new value.

Address: 0h base + 714h offset = 714h



PCS\_DIVE2 field descriptions

| Field            | Description                                                                                                      |
|------------------|------------------------------------------------------------------------------------------------------------------|
| 0–11<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                          |
| 12–31<br>DIVE    | Divider End Value<br>This is the clock divider end value for the clock ramp-down phase when switching from PLL1. |

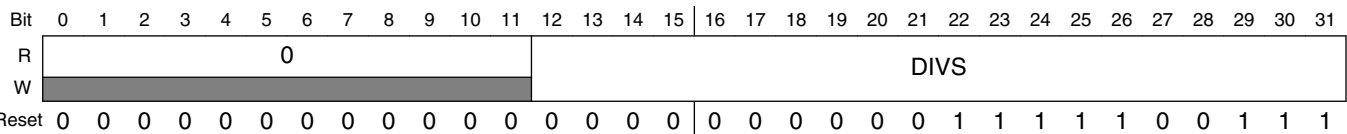
20.4.7 PCS Divider Start Register for PLL1 (PCS\_DIVS2)

This register defines the initial division value for the progressive system clock switching when switching the system clock source to PLL1 on ramp-up.

NOTE

Byte and half-word write accesses are not allowed for this register. Such an access will not result in an exception, but the value will not be loaded with the new value.

Address: 0h base + 718h offset = 718h



**PCS\_DIVS2 field descriptions**

| Field            | Description                                                                                                                 |
|------------------|-----------------------------------------------------------------------------------------------------------------------------|
| 0–11<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                     |
| 12–31<br>DIVS    | Divider Start Value<br><br>This is the start value of the clock divider for the clock ramp-up phase when switching to PLL1. |

**20.4.8 PCS Divider Change Register for PLL0 (PCS\_DIVC3)**

This register defines the rate of frequency change and initial change value for the progressive system clock switching when switching the system clock source to or from PLL0 on ramp-up and ramp-down, respectively.

**NOTE**

Byte write accesses are not allowed for the INIT field of this register. Such an access will not result in an exception, but the value will not be loaded with the new value.

Address: 0h base + 71Ch offset = 71Ch

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1  | 0  | 0  | 1  | 1  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**PCS\_DIVC3 field descriptions**

| Field             | Description                                                                                                                                                 |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>INIT      | Divider Change Initial Value<br><br>This is initial change value of the clock divider for the clock ramp-up phase when switching to PLL0.                   |
| 16–23<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                     |
| 24–31<br>RATE     | Divider Change Rate<br><br>This value controls the change value of the clock divider for the clock ramp-up and ramp-down phase when switching to/from PLL0. |

**20.4.9 PCS Divider End Register for PLL0 (PCS\_DIVE3)**

This register defines the final division value for the progressive system clock switching when switching the system clock source from PLL0 on ramp-down.

**NOTE**

Byte and half-word write accesses are not allowed for this register. Such an access will not result in an exception, but the value will not be loaded with the new value.

Address: 0h base + 720h offset = 720h

|       |   |   |   |   |   |   |   |   |   |   |    |    |      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12   | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    | DIVE |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 1  | 1  | 1  | 1  | 0  | 0  | 1  | 1  | 1  |

**PCS\_DIVE3 field descriptions**

| Field            | Description                                                                                                      |
|------------------|------------------------------------------------------------------------------------------------------------------|
| 0–11<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                          |
| 12–31<br>DIVE    | Divider End Value<br>This is the clock divider end value for the clock ramp-down phase when switching from PLL0. |

**20.4.10 PCS Divider Start Register for PLL0 (PCS\_DIVS3)**

This register defines the initial division value for the progressive system clock switching when switching the system clock source to PLL0 on ramp-up.

**NOTE**

Byte and half-word write accesses are not allowed for this register. Such an access will not result in an exception, but the value will not be loaded with the new value.

Address: 0h base + 724h offset = 724h

|       |   |   |   |   |   |   |   |   |   |   |    |    |      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12   | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    | DIVS |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 1  | 1  | 1  | 1  | 0  | 0  | 1  | 1  | 1  |

**PCS\_DIVS3 field descriptions**

| Field            | Description                                                                                                             |
|------------------|-------------------------------------------------------------------------------------------------------------------------|
| 0–11<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                 |
| 12–31<br>DIVS    | Divider Start Value<br>This is the start value of the clock divider for the clock ramp-up phase when switching to PLL0. |

## 20.5 Functional description

The progressive clock switch feature is enabled by setting SIU\_SYSDIV[PCSEN]. If PCSEN is set and a change to SIU\_SYSDIV[SYSCLKSEL] is detected by the PCS interface logic (see the PCS block diagram), then a progressive clock switch operation is triggered in the PCS block. The following rules govern the operation:

- If the switching is from the IRC clock, then a ramp-up command is sent to the PCS.
- If switching is to the IRC clock, then a ramp-down command is sent to the PCS.
- If the switching is between two clock sources and neither one is the IRC, then ramp-down and ramp-up commands are simultaneously sent to the PCS.

When the PCS block receives ramp-up and a ramp-down commands simultaneously, it executes the processes in this sequence:

1. Ramp-down process
2. Ramp-up process

During ramp-down, the rate of the frequency change is based on the PCS\_SDUR, PCS\_DIVC $n$ , and PCS\_DIVE $n$  registers, where  $n$  corresponds to the current system clock source selection. During ramp-up, the rate of the frequency change is based on the PCS\_SDUR, PCS\_DIVC $n$ , and PCS\_DIVS $n$  registers, where  $n$  corresponds to the target system clock source selection.

### NOTE

The progressive clock switching feature assumes that the IRC has the lowest clock frequency of the device. Do not attempt to use PCS when switching to or from a frequency that is less than than the IRC frequency.

When the PCS feature is enabled, the PCS module itself—not the SIU\_SYSDIV[SYSCLKSEL] field—controls the selection of the clock source.

While a PCS switching process is in progress:

- The clock source selected by the PCS may differ from the one requested by SIU\_SYSDIV[SYSCLKSEL].

- If software writes a new value to SIU\_SYSDIV[SYSCLKSEL] that corresponds to a clock source different from the one at the start of the switching process, the new value is ignored until the PCS process finishes.
- If software writes a SIU\_SYSDIV[SYSCLKSEL] value corresponding to the same clock source that was in effect at the start of the switching process, the PCS process aborts and returns to that original clock source.

At the end of the clock switching process:

- The SIU\_PCSIFR[PCSI] flag is asserted in the SIU. An interrupt to the CPU is generated if enabled by SIU\_PCSE[PCSE].
- The SIU\_PCSIFR[PCSMS] field indicates the current clock source selected by the PCS. Software can read PCSMS to learn whether or not the PCS switching process succeeded.

If the PCS switching fails (for example, due to incorrect programming), the clock source will not switch to a different one requested by SIU\_SYSDIV[SYSCLKSEL]. Instead, the clock source remains the one selected at the beginning of the attempted PCS switch.

### 20.5.1 Generic clock change requirements

For a maximum allowed change of the frequency ( $f_{\text{chg}}$ ) and for a given source clock frequency  $f_{\text{src}}$  (current or target clock source), the following equation gives the maximum allowed frequency change rate  $a_{\text{max}}$ .

$$a_{\text{max}} = \frac{f_{\text{chg}}}{f_{\text{src}}}$$

### 20.5.2 Configuration of PCS\_SDUR

The switch duration field PCS\_SDUR[SDUR] defines the duration of one system clock switch step in terms of the IRC clock period. After the expiration of this time, the module changes the clock divider value, which changes the frequency of the system clock.

### 20.5.3 Configuration of PCS\_DIVCn[RATE]

For a given maximum allowed frequency change rate  $a_{\text{max}}$ , the following table shows the value  $d$  based on the value programmed into the PCS\_DIVCn[RATE] register.

**Table 20-12. PCS\_DIVCn[RATE] values**

| $a_{\max}$ | $d$   | PCS_DIVCn[RATE] |
|------------|-------|-----------------|
| 0.05       | 0.012 | 12              |
| 0.10       | 0.048 | 48              |
| 0.15       | 0.112 | 112             |
| 0.20       | 0.184 | 184             |

### 20.5.4 Generic clock change properties

The following equation gives the number  $k$  of steps required to ramp down the source frequency ( $f_{\text{src}}$ ) to reach the IRC frequency ( $f_{\text{irc}}$ ).

$$k = \left\lceil 0.5 + \sqrt{0.25 - \frac{2 \left( 1 - \frac{f_{\text{src}}}{f_{\text{irc}}} \right)}{d}} \right\rceil$$

The following equation gives the number  $k$  of steps required to ramp up the IRC frequency ( $f_{\text{irc}}$ ) to reach the target frequency ( $f_{\text{targ}}$ ).

$$k = \left\lceil 0.5 + \sqrt{0.25 - \frac{2 \left( 1 - \frac{f_{\text{targ}}}{f_{\text{irc}}} \right)}{d}} \right\rceil$$

### 20.5.5 Clock ramp-down

The clock ramp-down:

- starts with the divider value 1 and with the given divider increment value PCS\_DIVCn[RATE].
- ends with the given divider value PCS\_DIVE<sub>n</sub>[DIVE].

The following equation gives the divider end value for clock ramp-down.

$$\text{PCS\_DIVE}_n[\text{DIVE}] = \frac{f_{\text{curr}}}{f_{\text{irc}}} \times 1000 - 1$$

In the equation,  $f_{\text{curr}}$  is the frequency of the currently selected system clock source, and  $f_{\text{irc}}$  is the frequency of the IRC clock.

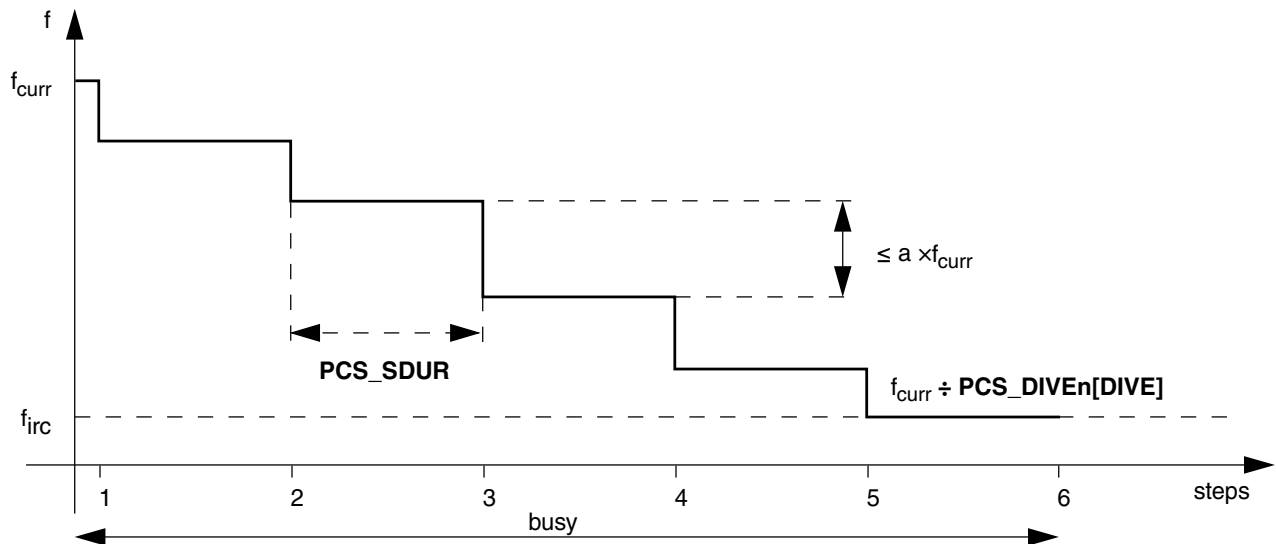


Figure 20-12. System clock ramp-down timing (k = 6 example)

## 20.5.6 Clock ramp-up

The clock ramp-up:

- starts with the given divider value  $\text{PCS\_DIVSn}[\text{DIVS}]$  and with the given divider decrement value  $\text{PCS\_DIVCn}[\text{INIT}]$ .
- ends with the divider value 1.

The following equation gives the initial divider change start value  $\text{PCS\_DIVCn}[\text{INIT}]$  for system clock ramp-up.

$$\text{PCS\_DIVCn}[\text{INIT}] = d \times k \times 1000$$

In the preceding equation,  $k$  is calculated from the second equation in [Generic clock change properties](#). The following equation gives the divider start value for clock ramp-up.

$$\text{PCS\_DIVSn}[\text{DIVS}] = \left(1 + d^{\frac{k(k+1)}{2}}\right) \times 1000 - 1$$



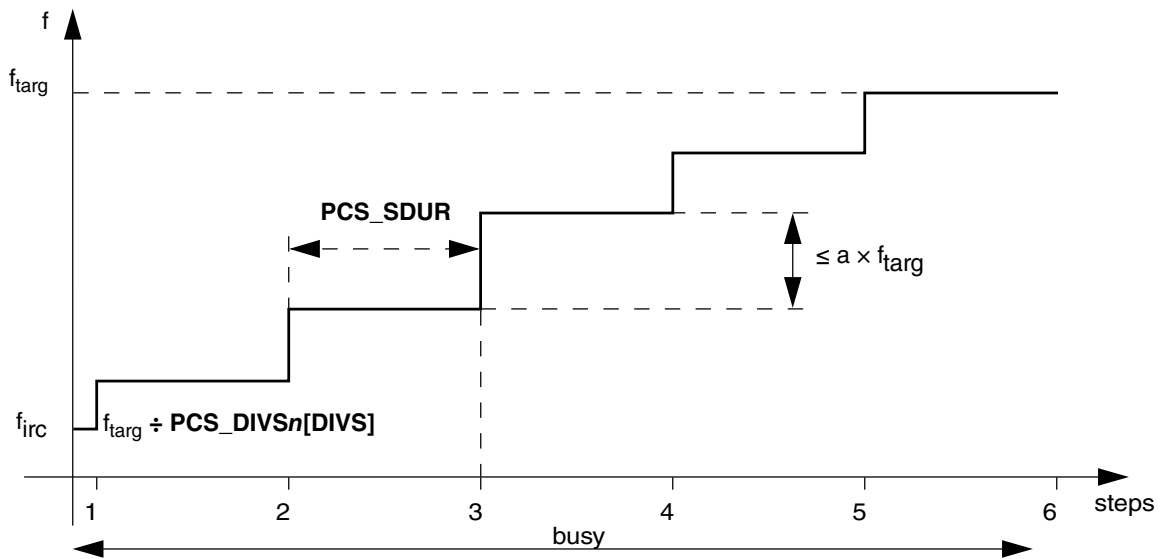


Figure 20-13. System clock ramp-up timing (k = 6 example)



# Chapter 21

## Dual PLL Digital Interface (PLLDIG)

### 21.1 Chip-specific PLLDIG information

The chip has one instance of the PLLDIG module. It includes two PLLs: PLL0 and PLL1.

For more chip-specific information, see the [Clocking](#) details, especially the [PLL](#) details.

### 21.2 Introduction

The chip provides a user interface and control over the dual PLL system composed of PLL0 and PLL1 analog blocks and the digital interface (PLLDIG). The two analog PLL blocks are cascaded, with the PHI1 output of PLL0 feeding the clock input of PLL1.

A key feature of the dual PLL architecture is the ability to drive peripherals from the PLL0 PHI output, which is non-modulated and independent of the core clock frequency. The core and platform clocks are driven by the PLL1 PHI frequency-modulated output.

### 21.3 Acronyms and Abbreviations

The following table contains sample acronyms and abbreviations used in this chapter.

**Table 21-1. Acronyms and abbreviated terms**

| Term  | Meaning                               |
|-------|---------------------------------------|
| PLL   | Phase Lock Loop                       |
| RFD   | Reduced Frequency Divider             |
| MFD   | Multiplication Factor Divider         |
| VCO   | Voltage Controlled Oscillator         |
| FMPLL | Frequency Modulated Phase Locked Loop |

## 21.4 Block Diagram

Please refer to the “Clocking chapter” of this *Reference Manual* to see the block diagram.

## 21.5 Features

The Dual PLLDIG has the following features:

- Supports dual PLL in cascaded mode with PLL0 clock out as reference clock to PLL1.
- Reference clock pre-divider for finer frequency synthesis resolution.
- Reduced frequency divider for decreasing the PLL0/PLL1 output clock frequency without causing the PLLs to lose lock.
- Programmable frequency modulation on PLL1.
- Lock detect circuitry reports when the PLLs have achieved frequency lock, and continuously monitors lock status to report loss of lock conditions.
  - User-selectable ability to generate an interrupt request upon loss of lock.
  - The loss of lock indication is sent to the FCCU via the system glue logic.

## 21.6 Modes of operation

The operating mode of the PLLs is indicated by the value of PLL $n$ CR[CLKCFG]. The various modes of operation for the PLL are:

- Normal mode with reference, with either PLL0 or both PLLs enabled
- PLL disabled

Normal mode is defined as the mode where the clocks are driven by the PLLs. When using a crystal for the clock reference, reset should remain asserted until the oscillator has stabilized.

### 21.6.1 Normal mode with reference, PLL0 or both PLLs enabled

In normal mode, PLL0 receives an input clock from the reference that can be prescaled by the pre-divider. PLL0 multiplies the frequency to create the PLL0 output clock. The user must supply a crystal that is within the appropriate frequency range, the crystal manufacturer's recommended external support circuitry and short signal route from the MCU to the crystal.

For information about the selection of a clock reference for each PLL, see [PLL](#) in the Clocking details.

PLL0 generates a non-modulated clock which drives PLL0 PHI1. PLL1 can generate a frequency modulated clock or a non-modulated clock (for example, locked on a single frequency). Write the PLL1FM register to program the modulation rate, modulation depth, and whether PLL modulating is enabled.

#### NOTE

PLL1 must be configured to work with a reference clock from PLL0 only after PLL0 has achieved lock.

For proper power sequencing of analog PLL blocks, write to PLL0CR[CLKCFG] before writing to PLL1CR[CLKCFG].

While powering down the PLLs: If PLL0:PHI1 is used as the source for PLL1, power down the PLLs in this sequence to ensure proper shutoff:

1. PLL1
2. PLL0

## 21.7 Memory map and register definition

This section provides the memory map and detailed descriptions of all registers for configuring the PLLs. The table below shows the memory map. Addresses are given as offsets from the module base address. All registers can be accessed using 8-bit, 16-bit or 32-bit addressing.

**PLLDIG memory map**

| Address offset (hex) | Register name                                | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|----------------------------------------------|-----------------|--------|-------------|----------------------------|
| 0                    | PLLDIG PLL0 Control Register (PLLDIG_PLL0CR) | 32              | R/W    | 0000_0000h  | <a href="#">21.7.1/794</a> |

*Table continues on the next page...*

### PLLDIG memory map (continued)

| Address offset (hex) | Register name                                             | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|-----------------------------------------------------------|-----------------|--------|-------------|----------------------------|
| 4                    | PLLDIG PLL0 Status Register (PLLDIG_PLL0SR)               | 32              | R/W    | 0000_0000h  | <a href="#">21.7.2/795</a> |
| 8                    | PLLDIG PLL0 Divider Register (PLLDIG_PLL0DV)              | 32              | R/W    | 2001_1008h  | <a href="#">21.7.3/796</a> |
| 20                   | PLLDIG PLL1 Control Register (PLLDIG_PLL1CR)              | 32              | R/W    | 0000_0000h  | <a href="#">21.7.4/798</a> |
| 24                   | PLLDIG PLL1 Status Register (PLLDIG_PLL1SR)               | 32              | R/W    | 0000_0000h  | <a href="#">21.7.5/799</a> |
| 28                   | PLLDIG PLL1 Divider Register (PLLDIG_PLL1DV)              | 32              | R/W    | 0001_0010h  | <a href="#">21.7.6/800</a> |
| 2C                   | PLLDIG PLL1 Frequency Modulation Register (PLLDIG_PLL1FM) | 32              | R/W    | 0000_0000h  | <a href="#">21.7.7/801</a> |
| 30                   | PLLDIG PLL1 Fractional Divide Register (PLLDIG_PLL1FD)    | 32              | R/W    | 0000_0000h  | <a href="#">21.7.8/802</a> |

## 21.7.1 PLLDIG PLL0 Control Register (PLLDIG\_PLL0CR)

Address: 0h base + 0h offset = 0h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |    |    |    |    |    |        |    |    |    |    |    |       |    |    |    |
|-------|----|----|----|----|----|----|--------|----|----|----|----|----|-------|----|----|----|
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22     | 23 | 24 | 25 | 26 | 27 | 28    | 29 | 30 | 31 |
| R     | 0  |    |    |    |    |    | CLKCFG |    | 0  |    |    |    | LOLIE | 0  |    |    |
| W     |    |    |    |    |    |    |        |    |    |    |    |    |       |    |    |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0      | 0  | 0  | 0  | 0  | 0  | 0     | 0  | 0  | 0  |

### PLLDIG\_PLL0CR field descriptions

| Field             | Description                                                                                                                                                    |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–21<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                        |
| 22–23<br>CLKCFG   | Clock Configuration<br>This field can be used to change the operating mode.<br><br>00 PLL off<br>01 Reserved<br>10 Reserved<br>11 Normal mode with PLL running |
| 24–27<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                        |
| 28<br>LOLIE       | Loss-of-lock interrupt enable                                                                                                                                  |

Table continues on the next page...

## PLLDIG\_PLL0CR field descriptions (continued)

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | <p>The LOLIE bit enables a loss-of-lock interrupt request when PLL0SR[LOLF] = 1. If PLL0SR[LOLF] = 0 or PLL0CR[LOLIE] = 0, the interrupt request is ignored. The interrupt request occurs only in normal mode, so the LOLIE bit has no effect if PLL is disabled.</p> <p><b>NOTE:</b> See <a href="#">PLL loss of lock interrupt</a> in the Clocking details for more information about configuring this field.</p> <p>0 Ignore loss-of-lock. Interrupt not requested.<br/>1 Enable interrupt request upon loss-of-lock.</p> |
| 29–31<br>Reserved | <p>This field is reserved.<br/>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                              |

## 21.7.2 PLLDIG PLL0 Status Register (PLLDIG\_PLL0SR)

Address: 0h base + 4h offset = 4h

|       |    |    |    |    |    |    |    |    |    |    |    |    |      |      |    |    |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|------|------|----|----|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12   | 13   | 14 | 15 |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |      |      |    |    |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |      |      |    |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0    | 0  | 0  |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28   | 29   | 30 | 31 |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    | LOLF | LOCK | 0  |    |
| W     |    |    |    |    |    |    |    |    |    |    |    |    | w1c  |      |    |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0    | 0  | 0  |

## PLLDIG\_PLL0SR field descriptions

| Field            | Description                                                                                     |
|------------------|-------------------------------------------------------------------------------------------------|
| 0–27<br>Reserved | <p>This field is reserved.<br/>This read-only field is reserved and always has the value 0.</p> |
| 28<br>LOLF       | Loss-of-lock flag                                                                               |

Table continues on the next page...

**PLLDIG\_PLL0SR field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | <p>This bit provides the interrupt request flag for the loss-of-lock condition. Software must write 1 to LOLF to clear it to 0; writing 0 has no effect. The flag is set to 1 when the PLL loses lock or when the divider/modulation registers are changed on the fly. This flag bit is sticky in the sense that if lock is reacquired, the bit remains set to 1 until cleared by either writing 1 or asserting reset. The LOLF is not asserted when the PLL is disabled, but disabling the PLL will not automatically clear the LOLF if it was previously asserted while the PLL was in normal mode.</p> <p>0 No loss of lock detected. Interrupt service not requested.<br/>1 Loss of lock detected. Interrupt service requested.</p> |
| 29<br>LOCK        | <p>Lock status bit</p> <p>Indicates whether PLL has acquired lock.</p> <p>0 PLL is unlocked.<br/>1 PLL is locked.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 30–31<br>Reserved | <p>This field is reserved.<br/>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

**21.7.3 PLLDIG\_PLL0 Divider Register (PLLDIG\_PLL0DV)**

The PLL0DV register provides the PHI/PHI1 output clock reduced frequency divider (RFDPHI and RFDPHI1), pre-divider (PREDIV), and loop divider (MFD). If this register's fields are modified without powering down the PLL, the PLL will lose lock and generate either an interrupt or a failure signal to the FCCU, based on which is enabled.

Address: 0h base + 8h offset = 8h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0 |   |   |   |   |   |   | 0 |   |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 1  |

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0  |    |    |    |    |    |    | 0  |    |    |    |    |    |    |    |    |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 0  |

**PLLDIG\_PLL0DV field descriptions**

| Field          | Description                                                                                                                        |
|----------------|------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>Reserved  | <p>This field is reserved.<br/>This read-only field is reserved and always has the value 0.</p>                                    |
| 1–4<br>RFDPHI1 | <p>PHI1 reduced frequency divider</p> <p>This 4-bit field determines the post-divider value for driving the PHI1 output clock.</p> |

*Table continues on the next page...*



## PLLDIG\_PLL0DV field descriptions (continued)

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | 0h to 3h    Reserved<br>4h        Divide by 4<br>5h        Divide by 5<br>...       ...<br>Eh        Divide by 14<br>Fh        Divide by 15                                                                                                                                                                                                                                                                                                 |
| 5–9<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                     |
| 10–15<br>RFDPHI   | PHI reduced frequency divider<br>This 6-bit field determines the VCO clock post-divider for driving the PHI output clock. See the equations in <a href="#">Clock configuration</a> .                                                                                                                                                                                                                                                        |
| 16<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                     |
| 17–19<br>PREDIV   | Input clock predivider<br>This 3-bit field controls the value of the divider on the input clock. The output of the predivider circuit generates the reference clock to the PLL analog loop. The PREDIV value 000b causes the input clock to be inhibited.<br>000    Clock inhibit<br>001    Divide by 1<br>010    Divide by 2<br>011    Divide by 3<br>100    Divide by 4<br>101    Divide by 5<br>110    Divide by 6<br>111    Divide by 7 |
| 20–24<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                     |
| 25–31<br>MFD      | Loop multiplication factor divider<br>This 7-bit field controls the value of the divider in the feedback loop. The value specified by the MFD bits establishes the multiplication factor applied to the reference frequency.<br>00h to 07h    Reserved<br>08h            Divide by 8<br>09h            Divide by 9<br>...            ...<br>7Eh            Divide by 126<br>7Fh            Divide by 127                                    |

## 21.7.4 PLLDIG PLL1 Control Register (PLLDIG\_PLL1CR)

Address: 0h base + 20h offset = 20h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |    |    |    |    |    |        |    |    |    |    |    |       |    |    |    |
|-------|----|----|----|----|----|----|--------|----|----|----|----|----|-------|----|----|----|
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22     | 23 | 24 | 25 | 26 | 27 | 28    | 29 | 30 | 31 |
| R     | 0  |    |    |    |    |    | CLKCFG |    | 0  |    |    |    | LOLIE | 0  |    |    |
| W     |    |    |    |    |    |    |        |    |    |    |    |    |       |    |    |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0      | 0  | 0  | 0  | 0  | 0  | 0     | 0  | 0  | 0  |

### PLLDIG\_PLL1CR field descriptions

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–21<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 22–23<br>CLKCFG   | Clock Configuration<br>This field can be used to change the operating mode.<br><br>00 PLL off<br>01 Reserved<br>10 Reserved<br>11 Normal mode with PLL running                                                                                                                                                                                                                                                                                                                                                                                            |
| 24–27<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 28<br>LOLIE       | Loss-of-lock interrupt enable<br><br>The LOLIE bit enables a loss-of-lock interrupt request when PLL1SR[LOLF] = 1. If PLL1SR[LOLF] = 0 or PLL1CR[LOLIE] = 0, the interrupt request is ignored. The interrupt request occurs only in normal mode, so the LOLIE bit has no effect if PLL is disabled.<br><br><b>NOTE:</b> See <a href="#">PLL loss of lock interrupt</a> in the Clocking details for more information about configuring this field.<br><br>0 Ignore loss-of-lock. Interrupt not requested.<br>1 Enable interrupt request upon loss-of-lock. |
| 29–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

## 21.7.5 PLLDIG PLL1 Status Register (PLLDIG\_PLL1SR)

Address: 0h base + 24h offset = 24h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |    |    |    |    |    |    |    |    |    |    |    |      |      |    |    |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|------|------|----|----|
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28   | 29   | 30 | 31 |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    | LOLF | LOCK | 0  |    |
| W     |    |    |    |    |    |    |    |    |    |    |    |    | w1c  |      |    |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0    | 0  | 0  |

**PLLDIG\_PLL1SR field descriptions**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–27<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 28<br>LOLF        | Loss-of-lock flag<br><br>This bit provides the interrupt request flag for the loss-of-lock condition. Software must write 1 to LOLF to clear it to 0; writing 0 has no effect. The flag is set to 1 when the PLL loses lock or when the divider/modulation registers are changed on the fly. This flag bit is sticky in the sense that if lock is reacquired, the bit remains set to 1 until cleared by either writing 1 or asserting reset. The LOLF is not asserted when the PLL is disabled, but disabling the PLL will not automatically clear the LOLF if it was previously asserted while the PLL was in normal mode.<br><br>0 No loss of lock detected. Interrupt service not requested.<br>1 Loss of lock detected. Interrupt service requested. |
| 29<br>LOCK        | Lock status bit<br><br>Indicates whether PLL has acquired lock.<br><br>0 PLL is unlocked.<br>1 PLL is locked.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 30–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

## 21.7.6 PLLDIG PLL1 Divider Register (PLLDIG\_PLL1DV)

The PLL1DV register provides the PHI output clock reduced frequency divider (RFDPHI) and loop divider (MFD). If this register's fields are modified without powering down the PLL, the PLL will lose lock and generate either an interrupt or a failure signal to the FCCU, based on which is enabled.

Address: 0h base + 28h offset = 28h

|       |   |   |   |   |   |   |   |   |   |   |        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |    |   |
|-------|---|---|---|---|---|---|---|---|---|---|--------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----|----|----|----|----|----|---|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10     | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25  | 26 | 27 | 28 | 29 | 30 | 31 |   |
| R     | 0 |   |   |   |   |   |   |   |   |   | RFDPHI |    |    |    |    |    | 0  |    |    |    |    |    |    |    |    | MFD |    |    |    |    |    |    |   |
| W     |   |   |   |   |   |   |   |   |   |   |        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |    |   |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0      | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0  | 0  | 1  | 0  | 0  | 0  | 0 |

### PLLDIG\_PLL1DV field descriptions

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–9<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                       |
| 10–15<br>RFDPHI   | PHI reduced frequency divider<br><br>This 6-bit field determines the VCO clock post-divider for driving the PHI output clock. See the equations in <a href="#">Clock configuration</a> .                                                                                                                                                                                                                                                                      |
| 16–24<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                       |
| 25–31<br>MFD      | Loop multiplication factor divider<br><br>This 7-bit field controls the value of the divider in the feedback loop. The value specified by the MFD bits establishes the multiplication factor applied to the reference frequency.<br><br><div> 00h to 09h    Reserved<br/> 10h           Divide by 16<br/> 11h           Divide by 17<br/> ...           ...<br/> 21h           Divide by 33<br/> 22h           Divide by 34<br/> 23h to 2Fh   Reserved </div> |

## 21.7.7 PLLDIG PLL1 Frequency Modulation Register (PLLDIG\_PLL1FM)

The PLL1FM register enables frequency modulation on PLL1 and provides controls for the FM spread spectrum, modulation depth, and modulation rate. This register should be written when PLL1CR[CLKCFG] = 00b (PLL1 powered down). After the PLL is on and has locked, changing the values of PLL1FM[MODEN] and PLL1FM[MODSEL] can result in the PLL losing its lock. The PLL would then require power cycling to regain normal functionality.

Address: 0h base + 2Ch offset = 2Ch

|       |    |        |        |        |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----|--------|--------|--------|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0  | 1      | 2      | 3      | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0  | MODEN  | MODSEL | MODPRD |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |    |        |        |        |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0  | 0      | 0      | 0      | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| Bit   | 16 | 17     | 18     | 19     | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0  | INCSTP |        |        |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |    |        |        |        |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0  | 0      | 0      | 0      | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**PLLDIG\_PLL1FM field descriptions**

| Field          | Description                                                                                                                                                                                                                                                                                              |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                  |
| 1<br>MODEN     | Modulation enable<br><br>This bit enables the frequency modulation.<br><br>0 Frequency modulation disabled.<br>1 Frequency modulation enabled.                                                                                                                                                           |
| 2<br>MODSEL    | Modulation selection<br><br>This bit selects whether modulation is centered around the nominal frequency or spread below the nominal frequency.<br><br>0 Center Spread modulation: Modulation centered around nominal frequency.<br>1 Down Spread modulation: Modulation spread below nominal frequency. |
| 3–15<br>MODPRD | Modulation period                                                                                                                                                                                                                                                                                        |

*Table continues on the next page...*

### PLLDIG\_PLL1FM field descriptions (continued)

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                 | $\text{MODPRD} = \text{round}\left(\frac{f_{\text{ref}}}{4 \times f_{\text{mod}}}\right)$ <p>MODPRD is the modulation period variable derived from the formula:<br/> where <math>f_{\text{ref}}</math> represents the frequency of the feedback divider and <math>f_{\text{mod}}</math> represents the modulation frequency.</p>                                                                                                                                                                                   |
| 16<br>Reserved  | <p>This field is reserved.<br/> This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 17–31<br>INCSTP | <p>Increment step</p> <p>This field is the INCSTP variable derived from the formula:<br/> <math display="block">\text{INCSTEP} = \text{round}\left(\frac{(2^{15} - 1) \times md \times \text{PLL1DV}[\text{MFD}]}{100 \times 5 \times \text{MODPRD}}\right)</math> where <math>md</math> represents the peak modulation depth in percentage (+/- for centered modulation, <math>-2 \times md</math> for modulation below nominal frequency) and MFD represents the nominal value of the feedback loop divider.</p> |

### 21.7.8 PLLDIG\_PLL1 Fractional Divide Register (PLLDIG\_PLL1FD)

The PLL1FD register provides the enable and fractional divide factor for the loop divider. This register should be written when PLL1CR[CLKCFG] = 00b (PLL1 powered down). After the PLL is running and has locked, changing the value of FDEN can result in the PLL losing its lock. The PLL would then require power cycling to regain normal functionality.

Address: 0h base + 30h offset = 30h

|       |    |      |    |    |    |    |    |    |    |    |    |    |    |    |         |         |
|-------|----|------|----|----|----|----|----|----|----|----|----|----|----|----|---------|---------|
| Bit   | 0  | 1    | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14      | 15      |
| R     | 0  |      |    |    |    |    |    |    | 0  |    |    |    |    |    |         |         |
| W     |    | FDEN |    |    |    |    |    |    |    |    |    |    |    |    | DTHDIS1 | DTHDIS0 |
| Reset | 0  | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0       | 0       |
| Bit   | 16 | 17   | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30      | 31      |
| R     |    |      | 0  |    |    |    |    |    |    |    |    |    |    |    |         |         |
| W     |    |      |    |    |    |    |    |    |    |    |    |    |    |    |         |         |
| Reset | 0  | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0       | 0       |

### PLLDIG\_PLL1FD field descriptions

| Field         | Description                                                                                      |
|---------------|--------------------------------------------------------------------------------------------------|
| 0<br>Reserved | <p>This field is reserved.<br/> This read-only field is reserved and always has the value 0.</p> |

Table continues on the next page...

**PLLDIG\_PLL1FD field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1<br>FDEN         | <p>Fractional Divide Enable</p> <p>This bit enables the fractional divider in the loop divider for PLL1.</p> <p>0 Fractional divide disabled.<br/>1 Fractional divide enabled.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 2–13<br>Reserved  | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 14<br>DTHDIS1     | <p>Rectangular Dither Disable</p> <p>Disables the Rectangular Phase/Frequency Detector dither.</p> <p>0 Dither enabled<br/>1 Dither disabled</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 15<br>DTHDIS0     | <p>Triangular Dither Disable</p> <p>Disables the Triangular Phase/Frequency Detector dither.</p> <p>0 Dither enabled<br/>1 Dither disabled</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 16–19<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 20–31<br>FRCDIV   | <p>Fractional divide input</p> <p>When the fractional divide is disabled, the VCO clock frequency is the product of the input clock and the loop divide factor (MFD). When fractional divide is enabled and dither is disabled (DTHDIS1 is 1), the VCO clock frequency is given by:</p> $\text{VCO Frequency} = \text{InputClock} \times (\text{PLL1DV}[\text{MFD}] + (\text{FRCDIV} \div 2^{12}))$ <p>When fractional divide is enabled and dither is enabled (DTHDIS1 is 0), the VCO clock frequency is given by:</p> $\text{VCO Frequency} = \text{InputClock} \times (\text{PLL1DV}[\text{MFD}] + (\text{FRCDIV} \div 2^{12}) + (1 \div 2^{13}))$ <p>In these equations, FRCDIV is the decimal equivalent of the FRCDIV field.</p> |

## 21.8 Functional description

This section explains the operation and configuration of the Dual PLLDIG module.

### 21.8.1 Input clock frequency

PLL0 and PLL1 are designed to operate over an input clock frequency range. The operating ranges for the PLLs are discussed in detail in the *Data Sheet*.

## 21.8.2 Clock configuration

The relationship between input and output frequency is determined by programming the PLL0DV and PLL1DV registers and calculated according to the following equations.

### NOTE

In the following equations, use the programmed value for each register field. For example, when PLL0DV[RFDPHI] is programmed to the value 1, use the value 1 in the equation for that field.

$$f_{\text{pll0\_phi}} = f_{\text{pll0\_ref}} \times \frac{\text{PLL0DV[MFD]}}{\text{PLL0DV[PREDIV]} \times \text{PLL0DV[RFDPHI]}}$$

**Figure 21-9. PLL0 PHI output frequency**

$$f_{\text{pll0\_phi1}} = f_{\text{pll0\_ref}} \times \frac{\text{PLL0DV[MFD]}}{\text{PLL0DV[PREDIV]} \times \text{PLL0DV[RFDPHI1]}}$$

**Figure 21-10. PLL0 PHI1 output frequency**

$$f_{\text{pll1\_phi}} = f_{\text{pll1\_ref}} \times \frac{\text{PLL1DV[MFD]}}{2 \times \text{PLL1DV[RFDPHI]}}$$

**Figure 21-11. PLL1 PHI output frequency (PLLDIG\_PLL1FD[FDEN] = 0)**

$$f_{\text{pll1\_phi}} = f_{\text{pll1\_ref}} \times \left( \frac{\text{PLL1DV[MFD]} + \frac{\text{PLL1FD[FRCDIV]}}{2^{12}} + \frac{1}{2^{13}}}{2 \times \text{PLL1DV[RFDPHI]}} \right)$$

**Figure 21-12. PLL1 PHI output frequency (PLLDIG\_PLL1FD[FDEN] = 1 and PLLDIG\_PLL1FD[DTHDIS1] = 0)**

### NOTE

When PLLDIG\_PLL1FD[FDEN] = 1 and PLLDIG\_PLL1FD[DTHDIS1] = 1, in [Figure 21-12](#) remove from the numerator: + 1/(2<sup>13</sup>)



$$f_{\text{pll1\_phi}} = \frac{f_{\text{pll0\_ref}} \times \text{PLL0DV}[\text{MFD}] \times \text{PLL1DV}[\text{MFD}]}{\text{PLL0DV}[\text{PREDIV}] \times \text{PLL0DV}[\text{RFDPHI1}] \times \text{PLL1DV}[\text{RFDPHI}] \times 2}$$

**Figure 21-13. PLL1 PHI output frequency when PLL0 PHI1 is PLL1 reference (PLLDIG\_PLL1FD[FDEN] = 0)**

$$f_{\text{pll1\_phi}} = \frac{f_{\text{pll0\_ref}} \times \text{PLL0DV}[\text{MFD}] \times (\text{PLL1DV}[\text{MFD}] + \frac{\text{PLL1FD}[\text{FRCDIV}]}{2^{12}} + \frac{1}{2^{13}})}{\text{PLL0DV}[\text{PREDIV}] \times \text{PLL0DV}[\text{RFDPHI1}] \times \text{PLL1DV}[\text{RFDPHI}] \times 2}$$

**Figure 21-14. PLL1 PHI output frequency when PLL0 PHI1 is PLL1 reference (PLLDIG\_PLL1FD[FDEN] = 1 and PLLDIG\_PLL1FD[DTHDIS1] = 0)**

### NOTE

When PLLDIG\_PLL1FD[FDEN] = 1 and PLLDIG\_PLL1FD[DTHDIS1] = 1, in [Figure 21-14](#) remove from the numerator: + 1/(2<sup>13</sup>)

The relationship between the VCO frequency ( $f_{\text{VCO}}$ ) and the output frequency of the PLLs is determined by the value of PLL1DV[RFDPHI], PLL0DV[RFDPHI], and PLL0DV[RFDPHI1], according to the following equations:

$$f_{\text{pll0\_VCO}} = f_{\text{pll0\_phi}} \times 2 \times \text{PLL0DV}[\text{RFDPHI}]$$

$$f_{\text{pll1\_VCO}} = f_{\text{pll1\_phi}} \times 2 \times \text{PLL1DV}[\text{RFDPHI}]$$

$$f_{\text{pll0\_VCO}} = f_{\text{pll0\_phi1}} \times 2 \times \text{PLL0DV}[\text{RFDPHI1}]$$

### NOTE

$f_{\text{pll0\_phi1}}$  is the reference clock generated by PLL0 for PLL1.

When programming the PLLs, user software must not violate the maximum system clock frequency or max/min VCO frequency specification PLL0 and PLL1. Furthermore, the PLL0DV[PREDIV] value must not be set to any value that causes the input frequency to the phase detector of analog PLL blocks to go below the prescribed ranges.

The PLL $n$ SR[LOCK] flag is immediately negated if the fields of PLL $n$ DV are changed without powering down the analog PLLs.

## NOTE

The PLL must be powered down and powered up by writing 00b to the PLL $n$ CR[CLKCFG] fields before PLL0DV[PREDIV], PLL0DV[MFD], PLL1DV[MFD], or the reference input clock values are modified.

The recommended procedure to program the PLLs and engage normal mode appears in [Initialization information](#).

### 21.8.3 Loss of lock

In the event that the PLL loses lock, the PLL digital interface registers provide the flexibility to select whether to generate an interrupt. If an interrupt is asserted, a failure event is also sent to the FCCU.

A loss-of-lock interrupt and FCCU failure event are generated only when the PLL is operating in normal mode. The PLL $n$ SR[LOLF] bits are not asserted when the PLLs are disabled (PLL $n$ CR[CLKCFG]=00b). If an LOLF flag was asserted while the PLL was in normal mode and then the PLL is disabled, the PLL disabling does not automatically clear the flag.

The lock indication from the analog PLL is synchronized and stored in the status register. When the analog PLL loses lock or regains lock, the status register indicates it immediately (after synchronization time).

### 21.8.4 Frequency modulation

Frequency modulation uses a triangular profile as shown in [Figure 21-15](#). The modulation frequency and depth are controlled using PLL1FM[MODPRD] and PLL1FM[INCSTP].

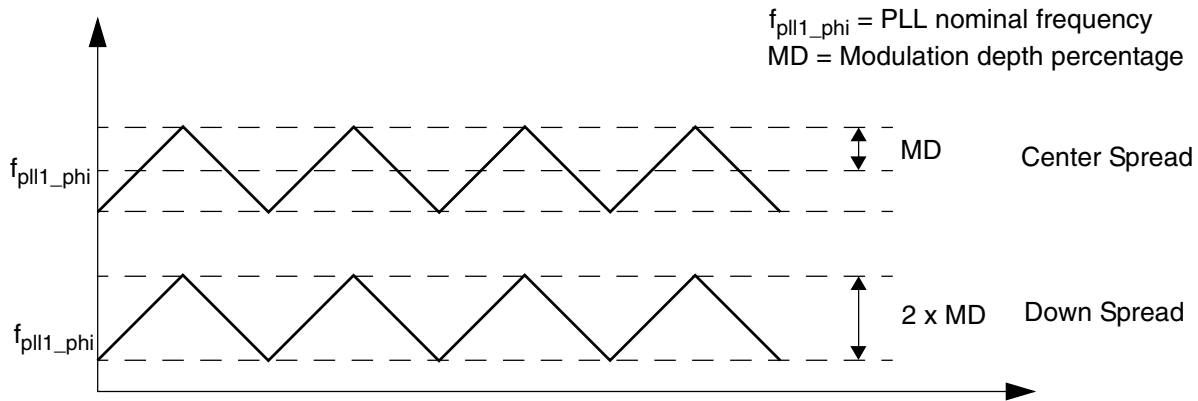


Figure 21-15. Triangular frequency modulation

**NOTE**

The device maximum operating frequency includes the frequency modulation. If center modulation is used, the  $f_{\text{sys}}$  must be below  $f_{\text{max}}$  by MD percentage such that  $f_{\text{max}} = f_{\text{sys}} \times (1 + (\text{MD}\% / 100))$ .

The following equations define how to calculate PLL1FM[MODPRD] and PLL1FM[INCSTP] based on the frequency of the feedback divider ( $f_{\text{ref}}$ ), the modulation frequency ( $f_{\text{mod}}$ ) and the modulation depth percentage (MD).

$$\text{PLL1FM}[\text{MODPRD}] = \text{round} \left( \frac{f_{\text{pll1\_ref}}}{4 \times f_{\text{mod}}} \right)$$

Figure 21-16. PLL1FM[MODPRD] calculation

The equation to determine PLLFM[INCSTP] is shown in [Figure 21-17](#).

$$\text{PLL1FM}[\text{INCSTP}] = \text{round} \left( \frac{(2^{15} - 1) \times \text{MD} \times \text{PLL1DV}[\text{MFD}]}{100 \times 5 \times \text{PLL1FM}[\text{MODPRD}]} \right)$$

Figure 21-17. PLL1FM[INCSTP] calculation

PLL1FM[MODPRD] and PLL1FM[INCSTP] are subject to the following restriction:

$$(\text{PLL1FM}[\text{MODPRD}] \times \text{PLL1FM}[\text{INCSTP}]) < 2^{15}$$

Figure 21-18. PLL1FM[MODPRD] and PLL1FM[INCSTP] restriction

Due to the preceding rounding operations, the effective modulation depth applied to the FMPLL is as shown in [Figure 21-19](#).

$$\text{ModulationDepth} = \text{round} \left( \frac{\text{PLL1FM}[\text{MODPRD}] \times \text{PLL1FM}[\text{INCSTP}] \times 100 \times 5}{(2^{15} - 1) \times \text{PLL1DV}[\text{MFD}]} \right)$$

**Figure 21-19. Modulation Depth**

As an example, suppose the following configuration, in which PLL1DV[RFDPHI] represents the programmed value of that field:

- Input frequency: 40 MHz
- Loop divider (PLL1DV[MFD]): 20
- VCO frequency: 40 MHz × 20 = 800 MHz
- PLL output frequency: 800 MHz / (2 × PLL1DV[RFDPHI])
- Center spread (PLL1FM[MODSEL] = 0)
- Modulation frequency: 100 kHz
- Modulation depth: +/- 2.0% (4% peak-to-peak)
- PLL1FM[MODPRD] = round[(40 × 10<sup>6</sup>) / (4 × 100 × 10<sup>3</sup>)] = round[100.00] = 100
- PLL1FM[INCSTP] = round[((2<sup>15</sup> - 1) × 2 × 20) / (100 × 5 × 100)] = round[26.2136] = 26
- PLL1FM[MODPRD] × PLL1FM[INCSTP] = 100 × 26 = 2600 (which is less than 2<sup>15</sup>)
- MD (quantized) = ((100 × 26 × 100 × 5) / ((2<sup>15</sup> - 1) × 20)) = 1.983703%

In this example, the modulation error is 0.0163%.

The recommended sequence for programming the FM is:

1. Power down PLL1 by writing PLL1CR[CLKCFG] = 00b.
2. Program PLL1FM while PLL1 is powered down.
3. Power up PLL1 by writing PLL1CR[CLKCFG] = 11b.

## 21.9 Maximum lock time

Maximum lock time is for the condition when the operating mode of the PLL has changed from power-down to Normal mode, input control bits are stable, and supplies (analog and digital) have attained levels shown in the *Data Sheet*. It is also the time in which the PLLs regain lock after a loss of lock event, assuming all inputs are stable and supplies are within specifications.

## 21.10 Initialization information

Coming out of reset PLL0 and PLL1 are disabled. The Dual PLLDIG initialization procedure is described in [PLLDIG initialization information](#) in the Clocking details.



## Chapter 22

# Clock Monitor Unit (CMU)

### 22.1 Chip-specific CMU information

The chip has nine instances of the CMU module. Each CMU module is programmed independently.

For more chip-specific information, see [Clock monitoring](#).

### 22.2 Introduction

The Clock Monitor Unit (CMU), also referred to as Clock Quality Checker or Clock Fault Detector, serves three purposes:

- Measures the frequency of clock source CLKMT0\_RMN with CLKMN0\_RMT as the reference clock
- Monitors CLKMN0\_RMT frequency with CLKMT0\_RMN as reference clock
- Monitors CLKMN1 frequency with CLKMT0\_RMN as reference clock and detects if the monitored clock frequency leaves an upper or lower frequency boundary

#### NOTE

See the "Clocking" chapter for chip-specific sources used by the CMU.

One of the tasks is to supervise the integrity of the various clock sources on the chip, for example CLKMN0\_RMT or CLKMN1. If the monitored clock frequency is less than the reference clock, or it violates an upper or lower frequency boundary, the CMU detects and reports this event. These events signal the FCCU, which can take the necessary corrective actions as its configuration dictates.

The CMU can monitor CLKMN0\_RMT, which must have a frequency higher than that of CLKMT0\_RMN divided by the factor shown in CMU\_CSR[RCDIV], and reports this event. The CMU can also monitor CLKMN1 and generate an event if CLKMN1 is greater than a high frequency boundary or less than a low frequency boundary. The upper and lower frequency boundaries are defined by the CMU High Frequency Reference Register (CMU\_HFREFR) and CMU Low Frequency Reference Register (CMU\_LFREFR).

The second task of the CMU is to provide a frequency meter, which allows measuring the frequency of a clock source against a reference clock. This is useful to correct/calculate the time deviation of a counter that is clocked by the metered clocks.

### **NOTE**

See the "Clocking" chapter for the number of CMU instances on this chip.

## **22.2.1 Main features**

- CLKMT0\_RMN frequency measurement with CLKMN0\_RMT as reference clock.
- CLKMN0\_RMT monitoring with respect to  $\text{CLKMT0\_RMN} \div 2^{\text{CSR[RCDIV]}}$  clock.
- Upper or lower frequency boundary monitoring of CLKMN1 with respect to  $\text{CLKMT0\_RMN} \div 4$ .
- Event generation for various failures detected inside monitoring unit.

## **22.3 Block diagram**

The block diagram of the CMU module(s) is shown in the "Clocking" chapter of this Reference Manual.

## **22.4 Signals**

The table below describes the signals on the boundary of the CMU (in alphabetical order).



**Table 22-1. Signal description**

| Signal     | I/O | Description                                                                                                                                                                                                                                                                                                  |
|------------|-----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CLKMNO_RMT | I   | Monitored Clock Signal 0/Metered Clock Signal Reference — Receives a clock signal that the CMU compares to a specified low-limit frequency to determine whether the frequency of the clock signal is greater than the specified limit. Also provides a reference clock signal for all metered clock signals. |
| CLKMN1     | I   | Monitored Clock Signal 1 — Receives a clock signal that the CMU compares to specified low-limit and high-limit frequencies to determine whether the frequency of the clock signal is between the specified limits.                                                                                           |
| CLKMT0_RMN | I   | Metered Clock Signal 0/Monitored Clock Signal Reference — Receives a clock signal that the CMU measures against a reference clock frequency. Also provides a reference clock signal for all monitored clock signals.                                                                                         |

**NOTE**

See the "Clocking" chapter for device specific clock sources of each CMU.

## 22.5 Register description and memory map

This section describes in address order all the CMU registers. Each description includes a standard register diagram with an associated figure number. The CMU memory map is listed in the following table.

**NOTE**

See "Clocking" chapter for register and field availability details.

**CMU memory map**

| Address offset (hex) | Register name                                             | Width (in bits) | Access | Reset value                 | Section/ page              |
|----------------------|-----------------------------------------------------------|-----------------|--------|-----------------------------|----------------------------|
| 0                    | CMU Control Status Register (CMU_CSR)                     | 32              | R/W    | <a href="#">See section</a> | <a href="#">22.5.1/814</a> |
| 4                    | CMU Frequency Display Register (CMU_FDR)                  | 32              | R      | 0000_0000h                  | <a href="#">22.5.2/815</a> |
| 8                    | CMU High Frequency Reference Register CLKMN1 (CMU_HFREFR) | 32              | R/W    | 0000_0FFFh                  | <a href="#">22.5.3/816</a> |
| C                    | CMU Low Frequency Reference Register CLKMN1 (CMU_LFREFR)  | 32              | R/W    | 0000_0000h                  | <a href="#">22.5.4/816</a> |
| 10                   | CMU Interrupt Status Register (CMU_ISR)                   | 32              | w1c    | <a href="#">See section</a> | <a href="#">22.5.5/817</a> |
| 18                   | CMU Measurement Duration Register (CMU_MDR)               | 32              | R/W    | 0000_0000h                  | <a href="#">22.5.6/818</a> |

## 22.5.1 CMU Control Status Register (CMU\_CSR)

Address: 0h base + 0h offset = 0h

|       |   |   |   |   |   |   |   |   |     |   |    |    |    |    |    |    |  |
|-------|---|---|---|---|---|---|---|---|-----|---|----|----|----|----|----|----|--|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8   | 9 | 10 | 11 | 12 | 13 | 14 | 15 |  |
| R     | 0 |   |   |   |   |   |   |   | SFM | 0 |    |    |    |    |    |    |  |
| W     |   |   |   |   |   |   |   |   |     |   |    |    |    |    |    |    |  |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0*  | 0 | 0  | 0  | 0  | 0  | 0  | 0  |  |

|       |    |    |    |    |    |    |        |    |    |    |    |    |    |    |       |    |     |
|-------|----|----|----|----|----|----|--------|----|----|----|----|----|----|----|-------|----|-----|
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22     | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30    | 31 |     |
| R     | 0  |    |    |    |    |    | CKSEL1 |    | 0  |    |    |    |    |    | RCDIV |    | CME |
| W     |    |    |    |    |    |    |        |    |    |    |    |    |    |    |       |    |     |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0*     | 0* | 0  | 0  | 0  | 0  | 0  | 0* | 0*    | 0  |     |

\* Notes:

- RCDIV field: Not all CMU blocks will utilize this feature. See the “Clocking” chapter for CMU implementation details.
- CKSEL1 field: Not all CMU blocks will utilize this feature. See the “Clocking” chapter for CMU implementation details.
- SFM field: Not all CMU blocks will utilize this feature. See the “Clocking” chapter for CMU implementation details.

### CMU\_CSR field descriptions

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–7<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                        |
| 8<br>SFM          | Start frequency measure.<br><br>The software can only set this bit to start a clock frequency measure. It is reset by hardware when the measure is ready in the CMU_FDR.<br><br><b>NOTE:</b> MDR[MD] must be written before enabling frequency measurement (CSR[SFM] = 1). Do not write MDR[MD] while CSR[SFM] = 1.<br><br>0 Frequency measurement is completed or not yet started<br>1 Frequency measurement is not completed |
| 9–21<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                        |
| 22–23<br>CKSEL1   | Frequency measure clock selection bit.<br><br>CKSEL1 selects the clock to be measured by the frequency meter. This only affects CMU instances that utilizes clock metering.<br><br>Not all CMU blocks will utilize this feature. See the “Clocking” chapter for device specific CMU implementation details.<br><br>00 CLKMT0_RMN is selected<br>01 Reserved<br>10 Reserved<br>11 CLKMT0_RMN is selected                        |
| 24–28<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                        |

Table continues on the next page...

**CMU\_CSR field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 29–30<br>RCDIV | <p>CLKMT0_RMN division factor.</p> <p>These bits specify the CLKMT0_RMN division factor. The output clock frequency is <math>f_{\text{CLKMT0\_RMN}} \div 2^{\text{CMU\_CSR[RCDIV]}}</math>. This output clock is used as reference clock to compare with CLKMN0_RMT for crystal clock monitor feature.</p> <p><b>NOTE:</b> Ensure clock source for CLKMN0_RMT is off before writing CMU_CSR[RCDIV].</p> <p>00 CLKMT0_RMN <math>\div</math> 1 (No division)<br/> 01 CLKMT0_RMN <math>\div</math> 2<br/> 10 CLKMT0_RMN <math>\div</math> 4<br/> 11 CLKMT0_RMN <math>\div</math> 8</p> |
| 31<br>CME      | <p>CLKMN1 monitor enable.</p> <p>0 CLKMN1 monitor is disabled<br/> 1 CLKMN1 monitor is enabled</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

**22.5.2 CMU Frequency Display Register (CMU\_FDR)**

The CMU\_FDR is used to determine the measured frequency of:

- CLKMT0\_RMN

with respect to the reference clock CLKMN0\_RMT.

**NOTE**

The CMU\_FDR should be read only when  
CMU\_CSR[SFM] = 0.

Address: 0h base + 4h offset = 4h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    | FD |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**CMU\_FDR field descriptions**

| Field            | Description                                                                                                                                                                                                                                                                                                                                         |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–11<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                  |
| 12–31<br>FD      | <p>Measured frequency bits.</p> <p>This register displays the measured frequency (<math>f_{\text{sel}}</math>) with respect to the reference clock (<math>f_{\text{CLKMN0\_RMT}}</math>). The measured value is given by the following formula:</p> $f_{\text{sel}} = (f_{\text{CLKMN0\_RMT}} \times \text{CMU\_MDR[MD]}) \div \text{CMU\_FDR[FD]}$ |

### 22.5.3 CMU High Frequency Reference Register CLKMN1 (CMU\_HFREFR)

The HFREFR is configured for the high frequency reference that the CMU will use for comparing against the monitored clock. The figure and table below show the CMU\_HFREFR register.

#### NOTE

For correct synchronization of CMU\_HFREFR write data, the user must ensure that a time interval of 10 peripheral bus clock cycles + 40 CLKMT0\_RMN clock cycles elapses before attempting subsequent writes to the same register.

Address: 0h base + 8h offset = 8h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16    | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | HFREF |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0     | 0  | 0  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  |    |

#### CMU\_HFREFR field descriptions

| Field            | Description                                                                                                                                                                                                   |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–19<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                       |
| 20–31<br>HFREF   | High Frequency reference value.<br>These bits determine the high reference value for the CLKMN1 frequency. The reference value is given by: $(\text{HFREF} \div 16) \times (f_{\text{CLKMT0\_RMN}} \div 4)$ . |

### 22.5.4 CMU Low Frequency Reference Register CLKMN1 (CMU\_LFREFR)

The LFREFR is configured for the low frequency reference that the CMU will use for comparing against the monitored clock. The figure and table below show the CMU\_LFREFR register.

#### NOTE

For correct synchronization of CMU\_LFREFR write data, the user must ensure that a time interval of 10 peripheral bus clock cycles + 40 CLKMT0\_RMN clock cycles elapses before attempting subsequent writes to the same register.

Address: 0h base + Ch offset = Ch

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16    | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | LFREF |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0     | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**CMU\_LFREFR field descriptions**

| Field            | Description                                                                                                                                                                                                        |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–19<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                            |
| 20–31<br>LFREF   | Low Frequency reference value.<br><br>These bits determine the low reference value for the CLKMN1 frequency. The reference value is given by:<br>$(\text{LFREF} \div 16) \times (f_{\text{CLKMT0\_RMN}} \div 4)$ . |

**22.5.5 CMU Interrupt Status Register (CMU\_ISR)****NOTE**

All flags in the CMU\_ISR are set asynchronously. This register must be read only after a CMU event is triggered. Otherwise, a read access on this register may fetch an incorrect value (see "Clocking" chapter for interrupt operation) .

**NOTE**

Before entering low-power stop modes, all clock frequency measurements in the CMU should be disabled. Failing to do so may result in ISR flags being spuriously set.

Address: 0h base + 10h offset = 10h

|       |          |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|----------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0        | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | Reserved |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |          |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |          |    |    |    |    |    |    |    |    |    |    |    |      |      |      |      |
|-------|----------|----|----|----|----|----|----|----|----|----|----|----|------|------|------|------|
| Bit   | 16       | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28   | 29   | 30   | 31   |
| R     | Reserved |    |    |    |    |    |    |    |    |    |    |    | FLCI | FHHI | FLLI | OLRI |
| W     |          |    |    |    |    |    |    |    |    |    |    |    | w1c  | w1c  | w1c  | w1c  |
| Reset | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0    | 0    | 0*   |

\* Notes:

- OLRI field: Not all CMU blocks will utilize this feature. See the "Clocking" chapter for specific CMU implementation details.

## CMU\_ISR field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–27<br>Reserved | This field is reserved.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 28<br>FLCI       | <p>CLKMN1 frequency less than reference clock event status.</p> <p>This bit is set by hardware when CLKMN1 frequency becomes lower than reference clock frequency (<math>f_{\text{CLKMTO\_RMN}} \div 4</math>) value and CLKMN1 is 'ON'. It can be cleared by software by writing '1'.</p> <p><b>NOTE:</b> This bit will show indeterministic behavior if the clock monitored by the CMU is not enabled .</p> <p>0 No FLC event<br/>1 FLC event occurred</p>                                                                                                                     |
| 29<br>FHHI       | <p>CLKMN1 frequency higher than high reference event status.</p> <p>This bit is set by hardware when CLKMN1 frequency becomes higher than the high frequency reference value determined by CMU_HFREFR[HFREF] and CLKMN1 is 'ON'. It can be cleared by software by writing '1'.</p> <p>0 No FHH event<br/>1 FHH event occurred</p>                                                                                                                                                                                                                                                |
| 30<br>FLLI       | <p>CLKMN1 frequency less than low reference event status.</p> <p>This bit is set by hardware when CLKMN1 frequency becomes lower than the low frequency reference value determined by CMU_LFREFR[LFREF], and CLKMN1 is 'ON'. Software clears this field by writing a '1'.</p> <p><b>NOTE:</b> This bit will show indeterministic behavior if the clock monitored by the CMU is not enabled .</p> <p>0 No FLL event<br/>1 FLL event occurred</p>                                                                                                                                  |
| 31<br>OLRI       | <p>Oscillator frequency less than <math>f_{\text{CLKMTO\_RMN}} \div 2^{\text{CMU\_CSR[RCDIV]}}</math> event status.</p> <p>This bit is set by hardware when the <math>f_{\text{CLKMN0\_RMT}}</math> is less than <math>f_{\text{CLKMTO\_RMN}} \div 2^{\text{CMU\_CSR[RCDIV]}}</math> frequency and CLKMN0_RMT is 'ON' . It can be cleared by software by writing '1'.</p> <p><b>NOTE:</b> While entering STOP mode, OLRI may be triggered due to stoppage of XOSC. To avoid this, disable the XOSC before entering STOP mode.</p> <p>0 No OLR event<br/>1 OLR event occurred</p> |

## 22.5.6 CMU Measurement Duration Register (CMU\_MDR)

Address: 0h base + 18h offset = 18h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    | MD |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

## CMU\_MDR field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                      |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–11<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                          |
| 12–31<br>MD      | Measurement duration bits<br><br>This field displays the measurement duration in terms of selected clock (CLKMT0_RMN) cycles. This value is loaded in the frequency meter down-counter. The down-counter starts counting when CMU_CSR[SFM] = 1.<br><br><b>NOTE:</b> Only values that do not cause a CMU_FDR[FD] overflow shall be written to CMU_MDR[MD]. There is no internal status of a CMU_FDR[FD] overflow. |

## 22.6 Functional description

### 22.6.1 Frequency meter

The purpose of the frequency meter is to evaluate the deviation from the nominal metered source (such as CLKMT0\_RMN) frequencies. Programming of the CMU\_CSR[CKSEL1] field is used to select one of the metered clocks from a multiplexer that drives a simple Frequency Meter (see the CMU Block Diagram in the "Clocking" chapter). The reference clock for the Frequency Meter is the CLKMN0\_RMT signal. The measurement starts when CMU\_CSR[SFM] = 1. The measurement duration is given by the contents of CMU\_MDR[MD] in terms of number of clock cycles of the selected metered clock. The CMU\_CSR[SFM] bit is cleared by hardware once the frequency measurement is complete and the count is loaded in CMU\_FDR[FD]. The frequency of the selected clock ( $f_{sel}$ ) can be derived from the value loaded in the CMU\_FDR[FD] as shown in the following equation:

$$f_{sel} = (f_{CLKMN0\_RMT} \times MDR[MD]) / FDR[FD]$$

### 22.6.2 CLKMN0\_RMT supervisor

If frequency of CLKMN0\_RMT is smaller than the frequency of CLKMT0\_RMN  $\div 2^{CMU\_CSR[RCDIV]}$  and CLKMN0\_RMT is 'ON', then:

- The CMU writes 1 to CMU\_ISR[OLRI].
- The CMU asserts the OLR signal.

**NOTE**

The equation below must be taken into consideration to guarantee correct  $f_{\text{CLKMN0\_RMT}}$  monitoring.

$$f_{\text{CLKMN0\_RMT}} \geq \frac{1.25 \times f_{\text{CLKMT0\_RMN}}}{2^{\text{CMU\_CSR[RCDIV]}}}$$

**Equation 1. Timing requirement****22.6.3 CLKMN1 supervisor**

The frequency of CLKMN1 ( $f_{\text{CLKMN1}}$ ) can be monitored by programming CMU\_CSR[CME] = 1. CLKMN1 monitoring starts as soon as CMU\_CSR[CME] = 1. This monitor can be disabled at any time by programming CMU\_CSR[CME] = 0.

If  $f_{\text{CLKMN1}}$  is greater than the reference value determined by fields CMU\_HFREFR[HFREF] and CLKMN1 is 'ON', then:

- The CMU writes 1 to CMU\_ISR[FHHI].
- The CMU asserts the FHH signal.

If  $f_{\text{CLKMN1}}$  is less than a reference clock frequency ( $f_{\text{CLKMT0\_RMN}} \div 4$ ) and the CLKMN1 is 'ON', then:

- The CMU writes 1 to CMU\_ISR[FLCI].
- The CMU asserts the FLC signal.

If  $f_{\text{CLKMN1}}$  is less than a reference value determined by the bits CMU\_LFREFR[LFREF] and the CLKMN1 is 'ON', then:

- The CMU writes 1 to CMU\_ISR[FLLI].
- The CMU asserts the FLL signal.

**Note**

$f_{\text{CLKMN1}}$  must be greater than  $f_{\text{CLKMT0\_RMN}} \div 4$  by at least 0.5 MHz in order to guarantee correct CLKMN1 monitoring.

An example of determining the  $\text{HFREF}_{\text{Actual}}$  is as follows. Assume a  $f_{\text{CLKMT0\_RMN}} = 16$  MHz with a accuracy of  $\pm 5\%$  (see *Data Sheet* for actual value). In order to monitor a  $f_{\text{CLKMN1}} = 200$  MHz, the ideal  $\text{HFREF}_{\text{Ideal}} = 800$ . The actual HFREF value will be 842 when the accuracy is taken into



consideration ( $\text{HFREF}_{\text{Actual}} = (\text{HFREF}_{\text{Ideal}} \div 0.95)$ ). The actual LFREF value will be 762 when accuracy is taken into consideration ( $\text{LFREF}_{\text{Actual}} = \text{LFREF}_{\text{Ideal}} \div 1.05$ ).



# Chapter 23

## Embedded Memories

### 23.1 Overview

This chip's features of embedded memory include:

- On-chip SRAM, including system SRAM and local instruction and data caches for each processor core
- On-chip system error correction code (ECC) flash memory
- End-to-end ECC (e2eECC) error detection and correction (not all modules support e2eECC)
- Embedded memories in select peripherals, including:
  - FlexCAN
  - M\_CAN
  - eDMA
  - FEC
  - eTPU
  - CSE

For more information about embedded memory in a peripheral, see the peripheral's dedicated chapter.

### 23.2 End-to-end Error Correction Code (e2eECC)

To support market requirements related to improved functional and transient fault detection capabilities, this chip includes end-to-end ECC (e2eECC) support. This e2eECC is structurally different from traditional "ECC at memory" functionality because

it provides robust error detection capabilities from one endpoint of an information transfer to another endpoint, with temporary information storage in one or more intermediate components.

Memory protected by ECC/EDC traditionally generates and checks additional error parity information local to the memory unit to detect and/or correct errors that have occurred on data stored in the memory. In contrast, e2eECC generates error protection codes at the source of data generation. It sends the encoded data and error protection codes to intermediate storage when a memory write is initiated by a bus master, and it performs a data integrity check using the previously stored error protection codes at a data memory when a read of stored information is requested by a bus master. Depending on the particular memory unit, the intermediate storage may transform the generated error protection codes into another format for storage and then regenerate the codes for provision when a request is made to read the stored information, or it may simply store the original protection codes unaltered.

To protect additional information associated with an access, the error protection codes are generated on the basis of more than the data associated with a storage location. In particular, address information corresponding to the access location of the stored information is combined with the stored data at the data source to generate error protection codes that cover certain types of addressing errors in the system interconnection or in the memory unit. The error protection codes may be checked at the memory unit on a store operation to ensure that no address information or data has been corrupted while the request has transitioned through the device from the bus master source. The memory unit may simply store the received data and protection codes at the address it receives, or it may check the error protection codes to ensure that address decoding within the storage memory was performed properly (although not all address decoding errors can be detected this way).

On a read request from a bus master, the memory unit retrieves the data information and error protection codes corresponding to the received address from the storage location or locations and supplies the data along with the error protection codes to the requesting device. The requesting device uses a locally stored address value corresponding to the read request to check the returning data and error protection codes to ensure that no errors have occurred in either addressing the memory or in the retrieved data. This approach ensures that the address sent for the request was not corrupted, the addressed location was actually accessed (to the extent it is possible to ensure), and the stored data was error free, to the extent the ECC coding scheme is able to detect. As a result, the fault coverage provided by the end-to-end check is considerably more robust than the previous implementation of locally generated and checked/corrected error protection at each memory unit.

The following table compares the traditional and e2eECC approaches.

**Table 23-1. ECC comparison of data write-then-read sequence**

| Traditional ECC                                                                                                                                                                                                                     | End-to-end ECC (e2eECC)                                                                                                                                                                                           |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Bus master initiates data write                                                                                                                                                                                                     | Bus master initiates data write and generates ECC checkbits based on 29-bit address and 64-bit data fields for the computational shell, or based on 30-bit address and 32-bit data field for the peripheral shell |
| Data write transfer routed from bus master to appropriate bus slave                                                                                                                                                                 | Data write transfer (including checkbits) routed from bus master to appropriate bus slave                                                                                                                         |
| For a bus memory slave, generate the ECC checkbits based on the data value and store data + checkbits into the memory                                                                                                               | For a bus memory slave, store data and checkbits into the memory                                                                                                                                                  |
| Bus master initiates data read of previously written memory location                                                                                                                                                                | Bus master initiates data read of previously written memory location                                                                                                                                              |
| Data read transfer routed from bus master to appropriate bus slave                                                                                                                                                                  | Data read transfer routed from bus master to appropriate bus slave                                                                                                                                                |
| For a bus memory slave, the memory array is accessed, the controller performs the ECC checkbit decode and syndrome generation, performs any needed single-bit correction, and drives the read data onto the system bus interconnect | For a bus memory slave, the memory array is accessed, and the controller passes the read data and associated checkbits onto the system bus interconnection                                                        |
| The bus master captures the read data and continues                                                                                                                                                                                 | The bus master captures the read data and associated checkbits, performs the ECC checkbit decode and syndrome generation, performs any needed single-bit correction, and continues                                |

The scope of differences in the operations "covered" by the ECC checks is readily apparent with the e2eECC concept, improving fault detection capabilities in two important aspects:

- The entire data transaction—from the initiating bus master, through the entire platform crossbar steering mechanism, to the destination slave target—is covered during write accesses. Likewise, a read is checked from the initiating bus master, through the crossbar steering mechanism, through the actual memory read and transmission of the data and checkbits back to the bus master, where the integrity and correctness of the entire transaction is checked.
- The selected ECC protects both the address field and the data field.

### NOTE

Compared to MPC5676R: For every SRAM and DMA transfer initiation, e2eECC on MPC5775E requires 2 additional clock cycles. The performance impact of these additional clocks can be reduced through the use of bursting.

In general, e2eECC improves implementation. The traditional ECC approach places the checkbit decode and error syndrome generation with the error/no\_error state, affecting whether the system bus transfer must be stalled (to correct a single bit error or report a

noncorrectable event) or is allowed to complete (for error-free transfers) in a critical timing arc that often sets the upper limit of the operating frequency of the microcontroller. With the e2eECC scheme, the checkbit decode and error syndrome logic is located in the requesting bus master with generally more degrees of implementation freedom; the error/no\_error state determination is removed from the system bus cycle termination logic, producing an improved timing arc and generally higher operating speeds.

Errors that occur within the system interconnect are typically manifested as an incorrect address, incorrect write data, or incorrect read data to be presented to the slave or back to the master. Errors occurring within the storage typically manifest themselves eventually in corrupted read data or checkbits being returned to a requester.

The following diagram provides an overview of the e2eECC implementation on this chip.

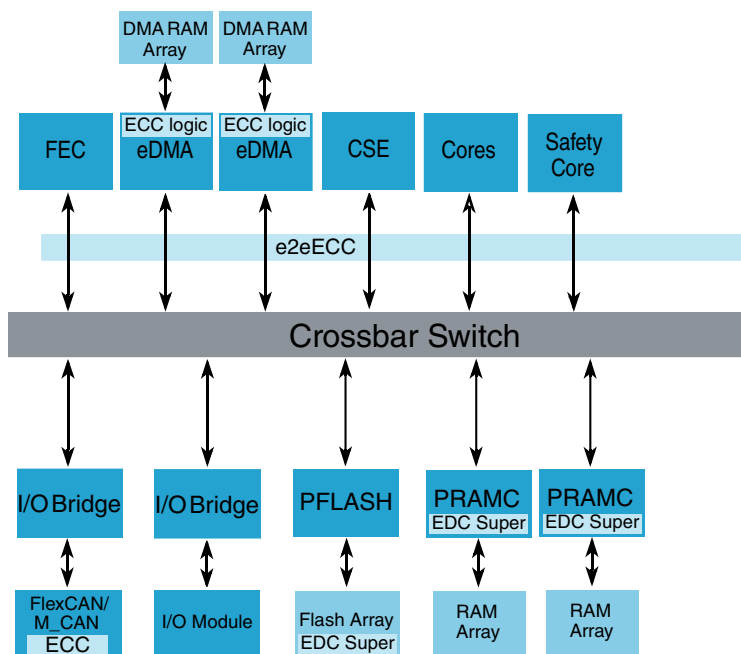


**Figure 23-1. e2eECC overview**

### NOTE

Due to the e2eECC implementation as it relates to the cores, treat ECC errors discovered during cache line fills as imprecise.

# Chapter 24

## RAM Controller (PRAMC)

### 24.1 Chip-specific PRAM information

This chip has two instances of the PRAM module.

#### 24.1.1 PRAM optional read wait states

The following table shows programming values for the BO\_DIS and FT\_DIS fields of the PRAM's PRCR1 register and the resulting access timing for a WRAP4 burst. For additional information, see [Optional read wait-state](#).

**Table 24-1. Read wait states for WRAP4 burst**

| BO_DIS | FT_DIS | WRAP4 timing |
|--------|--------|--------------|
| 0      | 0      | 1-1-1-1      |
| 0      | 1      | 2-1-1-1      |
| 1      | 0      | 1-1-1-1      |
| 1      | 1      | 2-2-2-2      |

#### 24.1.2 Reset effects on System RAM accesses

If any of the following types of reset event occurs during a read or write operation to System RAM, the completion of that access depends on the cycle at which the reset occurs. Data read from or written to System RAM before the reset event occurred is retained, but no other address locations are accessed or changed.

- External reset via RESET pin
- FCCU reset
- JTAG reset

- Software system reset
- Dual core reset

If the System RAM is cached, cache lines might retain indeterminate data that is not written to memory unless the region is set for writethrough mode.

### NOTE

Standby memory might contain the previous data values if a reset occurs while the cache is running in copyback mode.

For descriptions of reset types, see [Reset sources](#).

#### 24.1.2.1 Other reset types

The software external reset does not affect internal logic and therefore does not cause RAM corruption for any RAMs.

For all other reset types, no System RAM retention is guaranteed.

#### 24.1.2.2 Other RAMs

For all other RAMs on the chip, no retention is guaranteed for any reset type.

### 24.2 Introduction

This section provides an overview of the Platform RAM Controller. The RAM controller acts as an interface between the system bus (AHB-Lite 2.v6) and the integrated system RAM. It converts the protocols between the system bus and the dedicated RAM array interface.

The RAM controller supports two 64-bit AHB interfaces and a 64-bit RAM array interface. The primary AHB port provides a connection to the platform crossbar for direct RAM access from the various crossbar masters. Shown below in [Figure 24-1](#) is a simplified block diagram depicting the position of the RAM controller within a typical platform architecture.



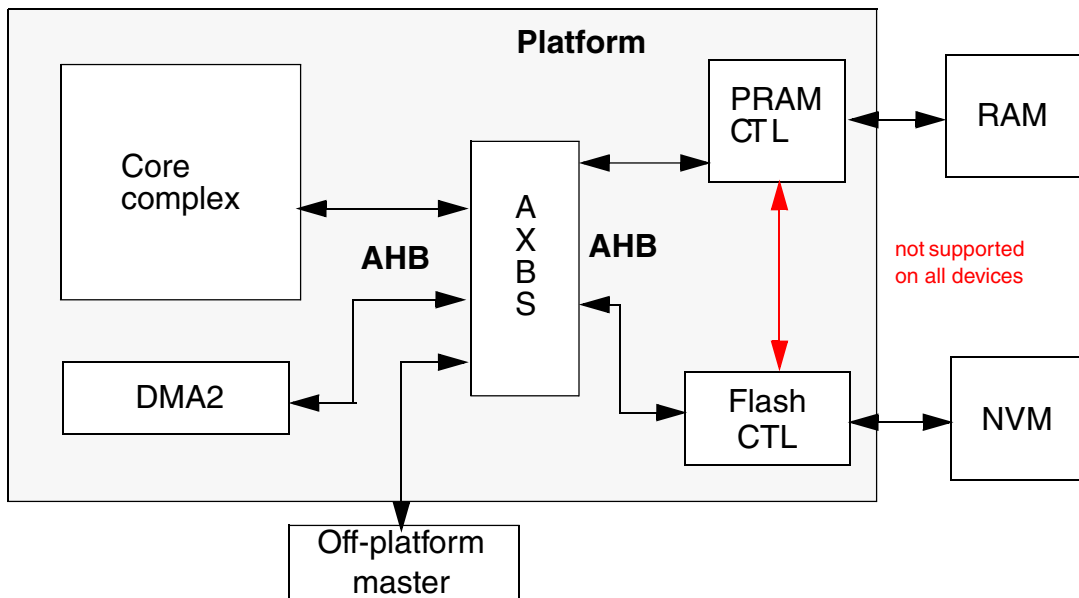


**Figure 24-1. Simplified platform block diagram**

The following list summarizes the key features of the RAM controller:

- System bus supports 64-bit data + 8-bit ECC AHB interfaces
- Array interface supports a 64-bit data + 8-bit ECC interface
- Late-write support via 64-bit data + 8-bit ECC storage buffer to support single-cycle write accesses
- Configurable read access timing (zero or one wait-state programmable) allowing use in large range of frequency targets
- Read-modify-write operation to support array write size less than a doubleword

The address path of the RAM controller contains a mux that chooses among the addresses presented on the system bus for a read request, either the address from the temporary holding register or the address from the late-write buffer. The temporary holding register contains the address which was presented during the AHB address phase of the write request. The request is stalled from being presented to the RAM by one cycle in order to present simultaneously the address and associated write data, which is not valid until the AHB dataphase. The late-write buffer holds write requests which are delayed to facilitate single-cycle response on the system bus.

The read datapath contains a mux that chooses the source of the read data to be presented on the system bus on a read request. In the event that a read request matches the contents of the late-write buffer, a RAM access is not required and the late-write buffer contents are selected onto the read databus. Otherwise, the read data is a result of a RAM access. Along with the read and write data, the corresponding ECC codewords traverse the entire datapath of the RAM controller, including the late-write storage buffer, to support end-to-end ECC (e2eECC) coverage.

The write datapath contains the mux that chooses the source of the write data as well as the control logic for generating read-modify-write operations on writes that are less than 64 bits in size. A write operation can be performed to empty the contents of the late-write buffer. In the case of back-to-back writes, the write may be forced to bypass the late-write buffer and instead be stored directly, precisely in the RAM.

### 24.3 SRAM controller memory map and register definition

The RAM controller module provides an IPS programming model mapped to a standard on-platform peripheral slot. The programming model can only be referenced using a 32-bit (word) access. Attempted references using different access sizes or to undefined (reserved) addresses generate an IPS error termination.

**NOTE**

Attempted updates to the PRAMC programming model while a PRAMC operation is in progress will result in non-deterministic behavior. Software must be architected to avoid this scenario by ensuring that PRAMC configuration changes are made only during system boot or when only one master is enabled. In multi-core devices, update the PRAMC configuration only when one core is active and no other masters, e.g., DMA or communications modules, are enabled. Move any instruction execution or memory references outside the system RAM while updating the PRAMC configuration, e.g., to the core local memory space.

**PRAMC memory map**

| Address offset (hex) | Register name                                       | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|-----------------------------------------------------|-----------------|--------|-------------|----------------------------|
| 0                    | Platform RAM Configuration Register 1 (PRAMC_PRCR1) | 32              | R/W    | 0000_0100h  | <a href="#">24.3.1/830</a> |

#### 24.3.1 Platform RAM Configuration Register 1 (PRAMC\_PRCR1)

The Platform RAM Configuration Register 1 (PRCR1) is used to specify operation of the RAM controller.

**NOTE**

This register is accessible only in supervisor mode. Accesses in user mode return a transfer error.

Address: 0h base + 0h offset = 0h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |    |    |    |    |    |          |    |          |                   |    |    |    |    |    |    |            |  |
|-------|----|----|----|----|----|----|----------|----|----------|-------------------|----|----|----|----|----|----|------------|--|
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22       | 23 | 24       | 25                | 26 | 27 | 28 | 29 | 30 | 31 |            |  |
| R     | 0  |    |    |    |    |    | Reserved |    | Reserved | P0_<br>BO_<br>DIS | 0  |    |    |    |    |    | FT_<br>DIS |  |
| W     |    |    |    |    |    |    | Reserved |    | Reserved | P0_<br>BO_<br>DIS |    |    |    |    |    |    | FT_<br>DIS |  |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0        | 1  | 0        | 0                 | 0  | 0  | 0  | 0  | 0  | 0  |            |  |

**PRAMC\_PRCR1 field descriptions**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–21<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 22–23<br>Reserved | This field is reserved.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 24<br>Reserved    | This field is reserved.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 25<br>P0_BO_DIS   | Port p0 read burst optimization disable.<br><br><b>NOTE:</b> The number of cycles taken for a RAM access can vary $\pm 1$ clock cycle depending on the RAM speed relative to the PRAMC controller clock frequency. If system RAM is running at the same frequency as the PRAMC controller, a random initial access takes 2 clock cycles. If system RAM is running at a slower frequency, a random initial access may take 3 clock cycles. Subsequent burst beats take either 1 or 2 cycles depending on RAM speed relative to the PRAMC controller clock frequency.<br><br>0 64-bit WRP4 read bursts are optimized such that the controller returns a 2-1-1-1 response when PRCR1[FT_DIS]=1<br>1 64-bit WRP4 read bursts are not optimized; the controller returns a 2-2-2-2 response when PRCR1[FT_DIS]=1 |
| 26–30<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 31<br>FT_DIS      | Flow through disabled.<br><br>This field defines the AHB response on reads. The state of this field has no impact on the response latency on writes. This bit is cleared by hardware reset.<br><br><b>NOTE:</b> Do not change the FT_DIS bit value while accessing system RAM. Relocate code programming the FT_DIS bit to another memory area, e.g., local core memory.<br><br>0 RAM read data is passed directly to the system bus, incurring no additional latency<br>1 RAM read data is registered prior to returning on the system bus, incurring 1 extra cycle of latency                                                                                                                                                                                                                            |

## 24.4 Functional description

This section describes the functions of the RAM controller.

### 24.4.1 Read and Write operations

The RAM controller processes read and write requests to on-chip RAM and provides a register interface for access to performance tuning functions.

#### NOTE

The RAM controller register interface is accessible only in supervisor mode. Accesses in user mode return a transfer error.

#### 24.4.1.1 Reads

Read transfers, of any size, can be configured to complete with a zero or one additional wait state response on the system bus.

#### 24.4.1.2 Optional read wait-state

The RAM controller can be optionally programmed to register RAM read data prior to returning it on the system bus. Setting the `PRCRx[FT_DIS]` field inserts a register in the read data path for use when operating the controller at high frequencies. The state of `PRCRx[FT_DIS]` field has no effect on writes.

A random, initial access will take 2 clock cycles (2T) to complete if `PRCRx[FT_DIS]=0`, and a WRAP4 burst will have an access time of 2-2-2-2. A random, initial access will take 3 clock cycles (3T) to complete if `PRCRx[FT_DIS]=1`, and a WRAP4 burst will have an access time of 3-2-2-2.

#### NOTE

The number of cycles taken for a RAM access can vary  $\pm 1$  clock cycle depending on the RAM speed relative to the PRAMC controller clock frequency. If system RAM is running at the same frequency as the PRAMC controller, a random initial access takes 2 clock cycles. If system RAM is running at a slower frequency, a random initial access may take 3 clock cycles. Subsequent burst beats take either 1 or 2 cycles

depending on RAM speed relative to the PRAMC controller clock frequency.

## 24.4.2 Writes

This section discusses various write operations of the RAM controller.

### 24.4.2.1 64-bit writes

Aligned 64-bit writes execute in a single AHB data phase cycle, allowing for zero wait states on back-to-back writes of these sizes. If, during the data phase of a write, a read is requested on the AHB, the write is registered in the late-write buffer, allowing the read to take place without a wait state. The valid buffered or late-write data is stored on the next available array address phase.

Back to back 64-bit writes execute slightly differently whereby the first write bypasses the late-write buffer. Rather, the write data is stored directly to the array in the same cycle in which it is valid on the AHB.

### 24.4.2.2 Less than 64-bit writes

Writes of size less than 64 bits incur a read-modify-write action as a consequence of the ECC coding scheme's 64-bit granularity. In the case of a read-modify-write action, the RAM controller performs SEC/DED on the read data. The write data is merged into the appropriate byte lanes along with the potentially corrected read data. A new codeword is generated based on the newly formed doubleword. The newly formed doubleword and its associated checkbits are subsequently written to the RAM. [Figure 24-3](#) provides details on the read-modify-write datapath. For details on the ECC coding scheme, refer to [Reliability considerations](#).

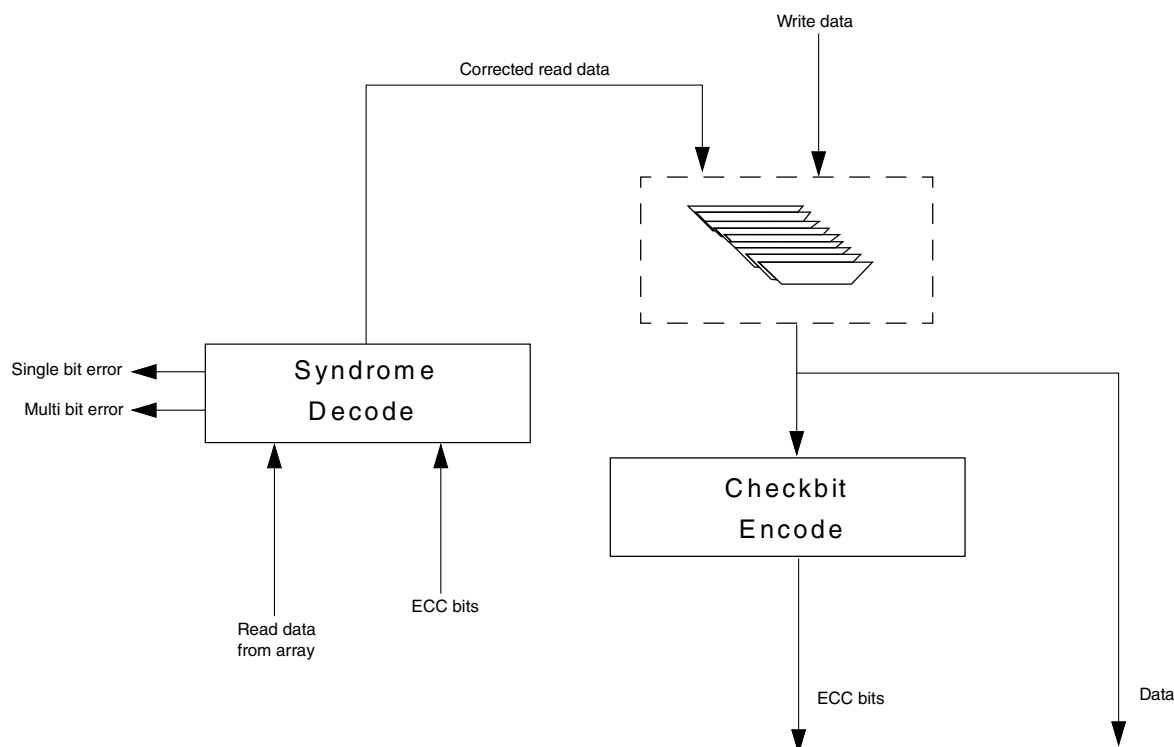


**Figure 24-3. Read-modify-write datapath**

On a read-modify-write operation, the array read data is registered after it is decoded and before it is merged with the AHB write data. Therefore, writes of size less than 64 bits require the insertion of one wait state before the data phase can be completed.

### 24.4.2.3 Unaligned writes

The RAM controller is compliant with the AMBA-AHB2.v6 Extensions specification with regard to byte strobes. The size of the transfer is sufficient to cover all bytes being written and covers more bytes in the case of an unaligned transfer. The address of the transfer is rounded down to the nearest boundary of the size of the transaction.

#### NOTE

Unaligned writes always generate read-modify-write operations in the RAM controller in order to preserve the validity of the ECC codeword.

## 24.5 Initialization/application information

It is essential that each memory address be written to a known value before it is read, to initialize the ECC. This includes reads generated from the read-modify-write operation which occurs when a write transfer of less than 64 bits or an unaligned write is requested. Without writing an address to a known value first, a read from this address will most likely generate an uncorrectable ECC event.

One possibility for initializing PRAMC memory space is to use a stored 64-bit word instruction such as Store Multiple Word (`e_stmw`) in Power Architecture VLE.

## 24.6 Reliability considerations

This section discusses reliability considerations of the RAM controller.

### 24.6.1 Hsiao ECC algorithm

The e2eECC scheme implements a single-error correction, double-error detection (SECDED) code using the so-called Hsiao odd-weight column criteria. These codes are named for M.Y. Hsiao, an IBM researcher who published extensively in the early 1970s on SECDED codes better suited for implementation in protecting (mainframe) computer memories than traditional Hamming codes.

The Hsiao codes are Hamming distance 4 implementations which provide the SECDED capabilities. The minimum odd-weight constraints defined by Hsiao are relatively simple in the resulting implementation of the parity check H matrix which defines the association between the data (and address) bits and the checkbits. They are:

1. There are no all-zeroes columns.
2. Every column is distinct.
3. Every column contains an odd number of ones, and hence is "odd weight".

In defining the H-matrix for this family of devices, these requirements from Hsiao were applied. Additionally, there are a variety of ECC codeword requirements associated with specific functional requirements associated with the system RAM that further dictated the specific column definitions. In any case, the resulting ECC is organized based on 64 data bits plus 29 address bits (the upper bits of the 32-bit address field minus the 3 bits which select the byte within the 64-bit (8-byte) data field).

| Checkbits [7:0] |    |    |    |    |    |    |    | Checkbits [7:0] |    |    |    |    |    |    |    | Checkbits [7:0] |    |    |    |    |    |    |    | Data Bit <sup>1</sup> |    |    |    |    |    |    |    |
|-----------------|----|----|----|----|----|----|----|-----------------|----|----|----|----|----|----|----|-----------------|----|----|----|----|----|----|----|-----------------------|----|----|----|----|----|----|----|
| XOR sum         |    |    |    |    |    |    |    | XOR sum         |    |    |    |    |    |    |    | XOR sum         |    |    |    |    |    |    |    | XOR sum               |    |    |    |    |    |    |    |
| Byte 7          |    |    |    |    |    |    |    | Byte 6          |    |    |    |    |    |    |    | Byte 5          |    |    |    |    |    |    |    | Byte 4                |    |    |    |    |    |    |    |
| 63              | 62 | 61 | 60 | 59 | 58 | 57 | 56 | 55              | 54 | 53 | 52 | 51 | 50 | 49 | 48 | 47              | 46 | 45 | 44 | 43 | 42 | 41 | 40 | 39                    | 38 | 37 | 36 | 35 | 34 | 33 | 32 |
| *               |    |    |    | *  |    |    | *  |                 |    |    |    | *  |    | *  | *  |                 | *  | *  | *  | *  |    |    |    | *                     | *  |    | *  |    | *  | *  | *  |
|                 | *  |    |    | *  |    |    |    | *               |    | *  |    | *  | *  |    | *  |                 | *  | *  |    | *  | *  | *  | *  | *                     |    | *  |    | *  |    | *  | *  |
|                 |    | *  |    | *  |    |    |    |                 |    | *  |    | *  | *  |    |    | *               | *  | *  |    | *  | *  | *  | *  | *                     |    | *  |    | *  |    | *  | *  |
|                 |    |    | *  | *  |    |    |    |                 | *  | *  | *  | *  | *  | *  | *  | *               | *  | *  | *  | *  | *  | *  | *  | *                     | *  | *  | *  | *  | *  | *  | *  |
|                 |    |    |    | *  | *  | *  | *  | *               | *  | *  | *  | *  | *  | *  | *  | *               | *  | *  | *  | *  | *  | *  | *  | *                     | *  | *  | *  | *  | *  | *  | *  |
|                 |    |    |    |    | *  | *  | *  | *               | *  | *  | *  | *  | *  | *  | *  | *               | *  | *  | *  | *  | *  | *  | *  | *                     | *  | *  | *  | *  | *  | *  | *  |
|                 |    |    |    |    |    | *  | *  | *               | *  | *  | *  | *  | *  | *  | *  | *               | *  | *  | *  | *  | *  | *  | *  | *                     | *  | *  | *  | *  | *  | *  | *  |
|                 |    |    |    |    |    |    | *  | *               | *  | *  | *  | *  | *  | *  | *  | *               | *  | *  | *  | *  | *  | *  | *  | *                     | *  | *  | *  | *  | *  | *  | *  |
|                 |    |    |    |    |    |    |    | *               | *  | *  | *  | *  | *  | *  | *  | *               | *  | *  | *  | *  | *  | *  | *  | *                     | *  | *  | *  | *  | *  | *  | *  |
|                 |    |    |    |    |    |    |    |                 | *  | *  | *  | *  | *  | *  | *  | *               | *  | *  | *  | *  | *  | *  | *  | *                     | *  | *  | *  | *  | *  | *  | *  |
|                 |    |    |    |    |    |    |    |                 |    | *  | *  | *  | *  | *  | *  | *               | *  | *  | *  | *  | *  | *  | *  | *                     | *  | *  | *  | *  | *  | *  | *  |
|                 |    |    |    |    |    |    |    |                 |    |    | *  | *  | *  | *  | *  | *               | *  | *  | *  | *  | *  | *  | *  | *                     | *  | *  | *  | *  | *  | *  | *  |
|                 |    |    |    |    |    |    |    |                 |    |    |    | *  | *  | *  | *  | *               | *  | *  | *  | *  | *  | *  | *  | *                     | *  | *  | *  | *  | *  | *  | *  |
|                 |    |    |    |    |    |    |    |                 |    |    |    |    | *  | *  | *  | *               | *  | *  | *  | *  | *  | *  | *  | *                     | *  | *  | *  | *  | *  | *  | *  |
|                 |    |    |    |    |    |    |    |                 |    |    |    |    |    | *  | *  | *               | *  | *  | *  | *  | *  | *  | *  | *                     | *  | *  | *  | *  | *  | *  | *  |
|                 |    |    |    |    |    |    |    |                 |    |    |    |    |    |    | *  | *               | *  | *  | *  | *  | *  | *  | *  | *                     | *  | *  | *  | *  | *  | *  | *  |
|                 |    |    |    |    |    |    |    |                 |    |    |    |    |    |    |    | *               | *  | *  | *  | *  | *  | *  | *  | *                     | *  | *  | *  | *  | *  | *  | *  |
|                 |    |    |    |    |    |    |    |                 |    |    |    |    |    |    |    |                 | *  | *  | *  | *  | *  | *  | *  | *                     | *  | *  | *  | *  | *  | *  | *  |
|                 |    |    |    |    |    |    |    |                 |    |    |    |    |    |    |    |                 |    | *  | *  | *  | *  | *  | *  | *                     | *  | *  | *  | *  | *  | *  | *  |
|                 |    |    |    |    |    |    |    |                 |    |    |    |    |    |    |    |                 |    |    | *  | *  | *  | *  | *  | *                     | *  | *  | *  | *  | *  | *  | *  |
|                 |    |    |    |    |    |    |    |                 |    |    |    |    |    |    |    |                 |    |    |    | *  | *  | *  | *  | *                     | *  | *  | *  | *  | *  | *  | *  |
|                 |    |    |    |    |    |    |    |                 |    |    |    |    |    |    |    |                 |    |    |    |    | *  | *  | *  | *                     | *  | *  | *  | *  | *  | *  | *  |
|                 |    |    |    |    |    |    |    |                 |    |    |    |    |    |    |    |                 |    |    |    |    |    | *  | *  | *                     | *  | *  | *  | *  | *  | *  | *  |
|                 |    |    |    |    |    |    |    |                 |    |    |    |    |    |    |    |                 |    |    |    |    |    |    | *  | *                     | *  | *  | *  | *  | *  | *  | *  |
|                 |    |    |    |    |    |    |    |                 |    |    |    |    |    |    |    |                 |    |    |    |    |    |    |    | *                     | *  | *  | *  | *  | *  | *  | *  |
|                 |    |    |    |    |    |    |    |                 |    |    |    |    |    |    |    |                 |    |    |    |    |    |    |    |                       | *  | *  | *  | *  | *  | *  | *  |
|                 |    |    |    |    |    |    |    |                 |    |    |    |    |    |    |    |                 |    |    |    |    |    |    |    |                       |    | *  | *  | *  | *  | *  | *  |
|                 |    |    |    |    |    |    |    |                 |    |    |    |    |    |    |    |                 |    |    |    |    |    |    |    |                       |    |    | *  | *  | *  | *  | *  |
|                 |    | </ |    |    |    |    |    |                 |    |    |    |    |    |    |    |                 |    |    |    |    |    |    |    |                       |    |    |    |    |    |    |    |

- NXP Semiconductors



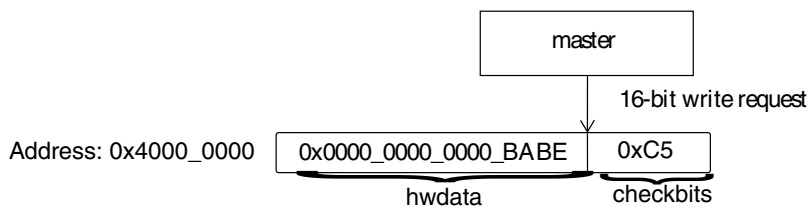
2. Bit numbering is AHB convention: bit 0 is LSB. D[7:0] corresponds to byte at address 0. D[63:56] corresponds to byte at address 7.

### 24.6.1.1 e2eECC considerations on less-than-64-bit write transactions

Recall writes of size less than 64-bit incur a read-modify-write action as a consequence of the ECC coding scheme 64-bit granularity. In the case of a read-modify-write action, the RAM controller performs SEC/DED on the read data. The RAM controller uses the following checkbit manipulation technique to ensure any faults in the PRAMC control logic associated with performing the read-modify-write action will ultimately be detectable by e2eECC:

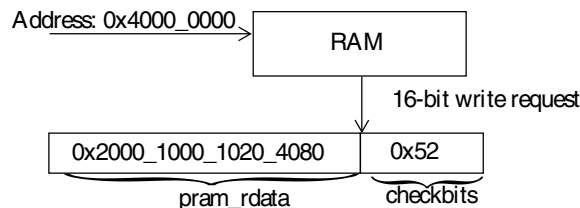
- On an 8-, 16-, or 32-bit write request, the write data is presented by the master on the AHB bus in the correct byte lanes, with the non-pertinent byte lanes zeroed out, and the associated checkbit is based on this "zero-padded" write data.

Consider the following example whereby the master presents a 16-bit write request at address 0x40000000:

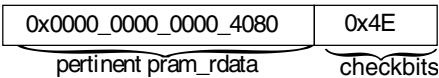


- The RAM controller detects that the request is a less-than-64-bit write request and initiates a read to address 0x40000000. An ECC check and potential single-bit correction is performed on the data returned from the RAM.

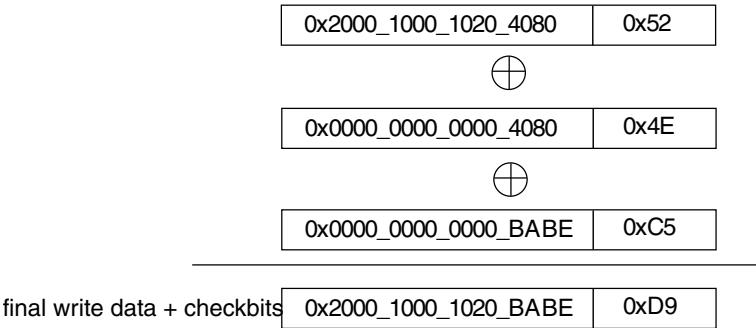
If a non-correctable ECC event is detected, the AHB request is terminated with error on the system bus.



- The RAM controller isolates the pertinent byte lanes of the read data by zero-padding the non-pertinent byte lanes and calculating the checkbit contribution associated with the read data in the byte lanes to be updated as a result of the write request.



- Merge the non-pertinent read data bytes with the write data in the appropriate byte lanes. Take the checkbit value that was stored in the RAM and remove the checkbit contribution associated with the read data in the byte lanes to be updated. Then introduce the checkbits associated with the write data. The result is ultimately a checkbit value associated with the complete 64-bit data to be written to the array.



This technique calculates the ultimate checkbit value on a read-modify-write transaction through a series of data manipulations, taking care to avoid discarding the original checkbits provided by the requesting master, such that faults in the crossbar (XBAR) and RAM controller datapath and control logic are covered by e2eECC.

Malfunction of the ECC logic described above may result in the corruption of the SEC/DED event reporting. The RAM controller performs EDC after ECC check on all read-modify-write transactions. If a mismatch is detected, indicating a failure in the ECC logic, the event is reported to the FCCU module.

24.6.2 Transaction monitor

The integrity of the address and data on a system RAM transaction is covered by e2eECC check performed by the requesting master. Fault detection coverage of the address path and control within the RAM controller is handled by a transaction monitor which verifies the integrity of the transactions between the RAM controller and the RAM array. The transaction monitor verifies RAM transactions initiated to service the following types of transactions:

- Direct system bus read or write request
- Read followed by write to fulfill a read-modify write transaction
- Write transaction to empty the late-write buffer

The function of the transaction monitor relies on a feedback path between the RAM controller and RAM array, wherein the RAM provides latched address and control feedback outputs as an indication of received inputs when a RAM access is initiated.

This feedback information is used by the RAM controller transaction monitor to verify the expected transaction. If a mismatch is detected, indicating a failure in the address generation or control logic within the RAM controller or the transmission path between the RAM controller and RAM array, the event is forwarded to the Fault Collection and Control Unit (FCCU).

Since writes to the RAM can be buffered, the transaction monitor also tracks the contents of the late-write buffer. When the late-write buffer is emptied, outputs from the RAM are evaluated against the expected contents of the late-write buffer as tracked by the transaction monitor.

As a countermeasure against false misses to the late-write buffer on reads, the transaction monitor also verifies the expected source of data returned on a read as supplied either from the RAM or from the late-write buffer.

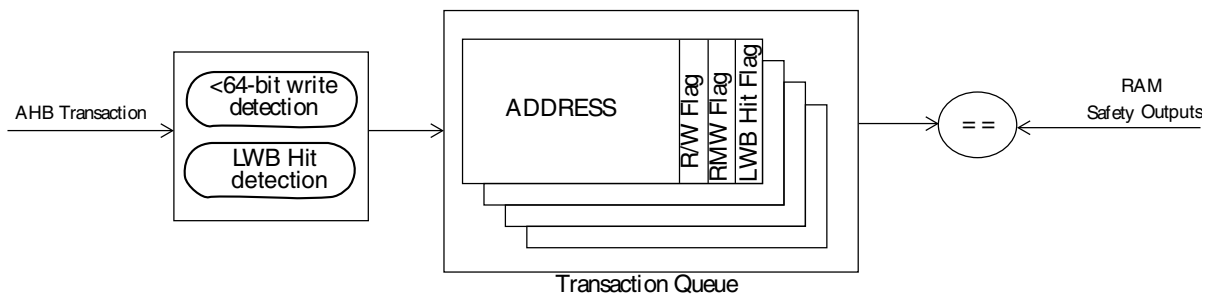


**Figure 24-4. RAM controller transaction monitor block diagram**



## Chapter 25

# Flash Memory Controller (PFLASH)

### 25.1 Chip-specific PFLASH information

The PFLASH registers PFCR1, PFCR2, and PFAPR control the interaction of master modules with the flash memory array by enabling/disabling prefetch or by controlling access to the array on a per-master basis.

Fields in these registers refer to the masters' logical ID numbers. For example, the PFAPR[M2AP] field controls flash memory array access by logical master 2, and the PFAPR[M3AP] field controls flash memory array access by logical master 3.

- For the list of bus masters' logical IDs, see [XBAR master and slave assignments](#).
- For logical master ID numbers that are unused on this chip, the corresponding PFLASH register fields are reserved.

For more information about configuring the PFLASH, see the [Platform Flash Configuration Register 3 \(PFLASH\\_PFCR3\)](#) details.

### 25.2 Introduction

The flash memory controller has these functions:

- It acts as an interface between the system bus (AHB-Lite 2.v6) and the flash memory array.

The flash memory controller supports two 64-bit AHB buses and a 256-bit read data interface from each flash memory array. Each AHB port contains a 4-entry, 4-way set-associative mini-cache as well as an associated controller that prefetches sequential lines of data from the flash arrays into the mini-cache. This buffer mechanism serves to deliver flash read data with zero-wait state response on lines that reside in the cache. AHB requests that miss the cache generate the needed flash array access and are forwarded to

the AHB upon completion. Each mini-cache entry is 256 bits in size, matching the flash array page size and providing 512 bytes of total buffered storage. Along with the read and write data, the corresponding ECC codewords traverse the entire datapath of the flash memory controller, including the mini-cache, to support end-to-end ECC (e2eECC) coverage.

## 25.3 Features

The following list summarizes the key features of the flash memory controller:

- Two 64-bit AHB interface ports (p0, p1) allowing simultaneous access to dedicated prefetch mini-cache per slave port.
- 256-bit read data bus + 64-bit write data bus
- Configurable read buffering and line prefetching support via 4-entry, 4-way set-associative mini-cache plus prefetch controller per AHB port to provide single-cycle "buffer hit" read response.
- Configurable access control based on read/write and AHB master ID attributes.
- Configurable access timing (wait-state programmable) allowing use in a wide range of frequency targets.
- Optional address pipelining capability to maximize throughput.
- Support for reporting of single- and multi-bit flash ECC events on a 64-bit doubleword boundary.

## 25.4 Block diagrams

[Figure 25-1](#) illustrates the connections of the flash memory controller and flash memory array within the chip.

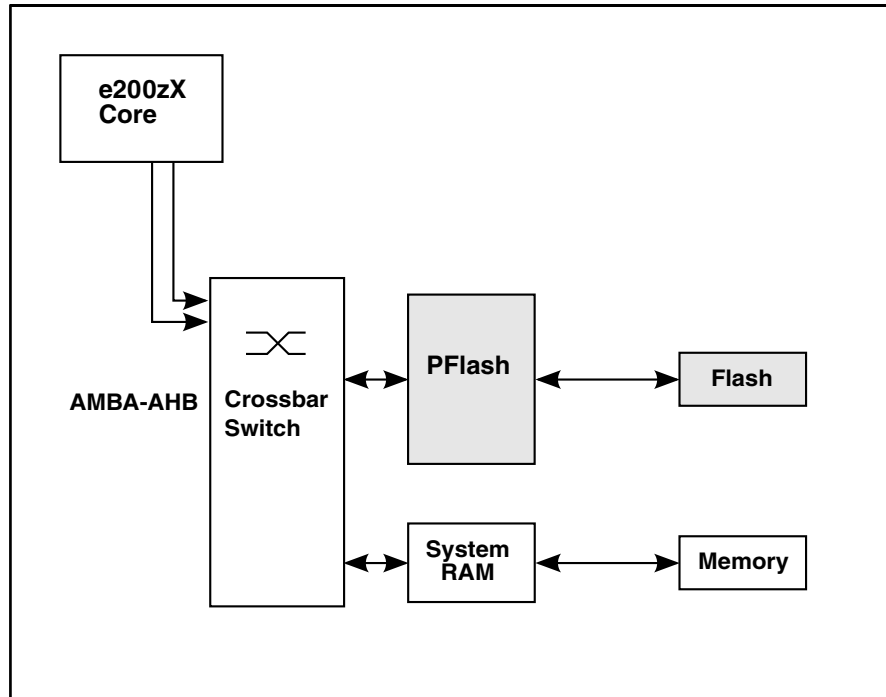


**Figure 25-1. Platform-centric simple block diagram with flash memory controller**

## 25.5 Flash memory controller memory map

The flash memory controller module provides an IPS programming model mapped to a standard 16 KB on-platform peripheral slot. The programming model consists of flash memory access configuration.

The programming model can only be referenced using a 32-bit (word) access. Attempted references using different access sizes or to undefined (reserved) addresses or in user mode generates an IPS error termination. The flash controller allows access to the programming model by all system bus masters.

The programming model can only be accessed in supervisor mode.

A write to any flash memory controller register, when executed concurrently with a flash memory access, will result in non-deterministic behavior and may include lockup of the flash memory controller. There is no idle indicator for the flash controller, so software must be architected to prevent any possible conflicts. The following recommendations will help.

- For multi-core devices, start only one core and execute initialization code to completion before starting the remaining cores.

- If flash must be reconfigured during application code execution, temporarily move all required code and data to system RAM before any writes to flash memory controller registers.
- If application software functionality includes invalidating a port read buffer, the recommended method for invalidating a port read buffer is:
  - a. Unlock the buffer's enable field by setting the PFLASH\_PFCR3[BFEN\_LK] field to '0'
  - b. Disable the buffer by setting it's associated enable field (PFLASH\_PFCR1[Pn\_BFEN]) to '0'
  - c. If needed, enable the buffer by setting it's associated enable field (PFLASH\_PFCR1[Pn\_BFEN]) to '1'

The above method requires updating flash memory controller registers, so software must ensure that no flash accesses are issued concurrently with any of the steps. In cases where that cannot be guaranteed, there is a workaround but its use is not generally recommended because it relies on a hardware side effect instead of the flash memory controller's built-in functions intended for invalidating port read buffers. The workaround is to issue an interlock write without an accompanying program or erase operation—an interlock write, without an accompanying program or erase, does not change any flash content.

### NOTE

1. If this workaround is implemented in application code, note that an interlock write invalidates only the buffers associated with the system bus port on which the interlock write was issued.
2. See the flash memory chapter for the steps to issue an interlock write.
3. See the flash memory controller's chip-specific information for any restrictions on the interlock write data size.

### PFLASH memory map

| Address offset (hex) | Register name                                            | Width (in bits) | Access | Reset value | Section/page               |
|----------------------|----------------------------------------------------------|-----------------|--------|-------------|----------------------------|
| 0                    | Platform Flash Configuration Register 1 (PFLASH_PFCR1)   | 32              | R/W    | 0000_0601h  | <a href="#">25.5.1/845</a> |
| 4                    | Platform Flash Configuration Register 2 (PFLASH_PFCR2)   | 32              | R/W    | 0000_0001h  | <a href="#">25.5.2/848</a> |
| 8                    | Platform Flash Configuration Register 3 (PFLASH_PFCR3)   | 32              | R/W    | 0000_0000h  | <a href="#">25.5.3/851</a> |
| C                    | Platform Flash Access Protection Register (PFLASH_PFAPR) | 32              | R/W    | FFFF_FFFFh  | <a href="#">25.5.4/853</a> |



### 25.5.1 Platform Flash Configuration Register 1 (PFLASH\_PFCR1)

The PFlash Configuration Register 1 (PFCR1) controls the operation of Port p0 of the PFLASH\_C55FM flash memory controller.

#### NOTE

See the chip-specific platform flash controller information for information on the masters.

Address: 0h base + 0h offset = 0h

| Bit   | 0         | 1         | 2         | 3         | 4         | 5         | 6        | 7        | 8        | 9        | 10       | 11       | 12       | 13       | 14       | 15       |
|-------|-----------|-----------|-----------|-----------|-----------|-----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| R     | P0_M15PFE | P0_M14PFE | P0_M13PFE | P0_M12PFE | P0_M11PFE | P0_M10PFE | P0_M9PFE | P0_M8PFE | P0_M7PFE | P0_M6PFE | P0_M5PFE | P0_M4PFE | P0_M3PFE | P0_M2PFE | P0_M1PFE | P0_M0PFE |
| W     |           |           |           |           |           |           |          |          |          |          |          |          |          |          |          |          |
| Reset | 0         | 0         | 0         | 0         | 0         | 0         | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        |

| Bit   | 16  | 17 | 18 | 19 | 20   | 21 | 22 | 23 | 24 | 25       | 26 | 27       | 28 | 29       | 30 | 31      |
|-------|-----|----|----|----|------|----|----|----|----|----------|----|----------|----|----------|----|---------|
| R     | APC |    |    | 0  | RWSC |    |    |    | 0  | P0_DPFEN | 0  | P0_IPFEN | 0  | P0_PFLIM |    | P0_BFEN |
| W     |     |    |    |    |      |    |    |    |    |          |    |          |    |          |    |         |
| Reset | 0   | 0  | 0  | 0  | 0    | 1  | 1  | 0  | 0  | 0        | 0  | 0        | 0  | 0        | 0  | 1       |

#### PFLASH\_PFCR1 field descriptions

| Field          | Description                                                                                                                                                                                                        |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>P0_M15PFE | Port0 Master 15 Prefetch enable. This bit controls whether prefetching may be triggered by AHB master 15.<br><br>0 No prefetching may be triggered by this master<br>1 Prefetching may be triggered by this master |
| 1<br>P0_M14PFE | Port0 Master 14 Prefetch enable. This bit controls whether prefetching may be triggered by AHB master 14.<br><br>0 No prefetching may be triggered by this master<br>1 Prefetching may be triggered by this master |
| 2<br>P0_M13PFE | Port0 Master 13 Prefetch enable. This bit controls whether prefetching may be triggered by AHB master 13.<br><br>0 No prefetching may be triggered by this master<br>1 Prefetching may be triggered by this master |
| 3<br>P0_M12PFE | Port0 Master 12 Prefetch enable. This bit controls whether prefetching may be triggered by AHB master 12.                                                                                                          |

Table continues on the next page...

**PFLASH\_PFCR1 field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                        |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | 0 No prefetching may be triggered by this master<br>1 Prefetching may be triggered by this master                                                                                                                  |
| 4<br>P0_M11PFE | Port0 Master 11 Prefetch enable. This bit controls whether prefetching may be triggered by AHB master 11.<br><br>0 No prefetching may be triggered by this master<br>1 Prefetching may be triggered by this master |
| 5<br>P0_M10PFE | Port0 Master 10 Prefetch enable. This bit controls whether prefetching may be triggered by AHB master 10.<br><br>0 No prefetching may be triggered by this master<br>1 Prefetching may be triggered by this master |
| 6<br>P0_M9PFE  | Port0 Master 9 Prefetch enable. This bit controls whether prefetching may be triggered by AHB master 9.<br><br>0 No prefetching may be triggered by this master<br>1 Prefetching may be triggered by this master   |
| 7<br>P0_M8PFE  | Port0 Master 8 Prefetch enable. This bit controls whether prefetching may be triggered by AHB master 8.<br><br>0 No prefetching may be triggered by this master<br>1 Prefetching may be triggered by this master   |
| 8<br>P0_M7PFE  | Port0 Master 7 Prefetch enable. This bit controls whether prefetching may be triggered by AHB master 7.<br><br>0 No prefetching may be triggered by this master<br>1 Prefetching may be triggered by this master   |
| 9<br>P0_M6PFE  | Port0 Master 6 Prefetch enable. This bit controls whether prefetching may be triggered by AHB master 6.<br><br>0 No prefetching may be triggered by this master<br>1 Prefetching may be triggered by this master   |
| 10<br>P0_M5PFE | Port0 Master 5 Prefetch enable. This bit controls whether prefetching may be triggered by AHB master 5.<br><br>0 No prefetching may be triggered by this master<br>1 Prefetching may be triggered by this master   |
| 11<br>P0_M4PFE | Port0 Master 4 Prefetch enable. This bit controls whether prefetching may be triggered by AHB master 4.<br><br>0 No prefetching may be triggered by this master<br>1 Prefetching may be triggered by this master   |
| 12<br>P0_M3PFE | Port0 Master 3 Prefetch enable. This bit controls whether prefetching may be triggered by AHB master 3.<br><br>0 No prefetching may be triggered by this master<br>1 Prefetching may be triggered by this master   |
| 13<br>P0_M2PFE | Port0 Master 2 Prefetch enable. This bit controls whether prefetching may be triggered by AHB master 2.<br><br>0 No prefetching may be triggered by this master<br>1 Prefetching may be triggered by this master   |
| 14<br>P0_M1PFE | Port0 Master 1 Prefetch enable. This bit controls whether prefetching may be triggered by AHB master 1.<br><br>0 No prefetching may be triggered by this master<br>1 Prefetching may be triggered by this master   |

*Table continues on the next page...*

**PFLASH\_PFCR1 field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 15<br>P0_MOPFE | Port0 Master 0 Prefetch enable. This bit controls whether prefetching may be triggered by AHB master 0.<br><br>0 No prefetching may be triggered by this master<br>1 Prefetching may be triggered by this master                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 16–18<br>APC   | Address Pipeline Control. This field controls the number of cycles that a subsequent flash read from the opposite AHB port can be initiated prior to the previous read data being valid.<br><br>This field applies to the configuration of Port0 and Port1.<br><br>For valid RWSC and APC combinations please refer to the device data sheet.<br><br><b>NOTE:</b> <ol style="list-style-type: none"> <li>1. A value of '1' in the most significant bit causes the flash controller to function in retrograde/legacy mode, in which there is no pipelining between the sampling of flash read data and the presentation of the next address for flash lookup.</li> <li>2. Flash operation is not guaranteed for RWSC/APC combinations other than those specified in the data sheet.</li> </ol><br>000 Pipelined access to the flash disabled.<br>001 A pipelined access can be initiated 1 cycle before the previous data is valid<br>010 A pipelined access can be initiated 2 cycles before the previous data is valid<br>011 A pipelined access can be initiated 3 cycles before the previous data is valid<br>1xx Pipelined access to the flash is disabled and one wait state is inserted before a subsequent access can be initiated |
| 19<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 20–23<br>RWSC  | Read Wait State Control. This field controls the number of wait-states to be added to the best-case flash array access time for reads. The best-case flash array access time for reads is one cycle.<br><br>This field must be set to a value corresponding to the operating frequency of the flash memory controller and the actual read access time of the flash memory controller.<br><br>For valid RWSC and APC combinations please refer to the device data sheet.<br><br><b>NOTE:</b> <ol style="list-style-type: none"> <li>1. Higher operating frequencies require non-zero settings for this field for proper flash operation.</li> <li>2. Flash operation is not guaranteed for RWSC/APC combinations other than those specified in the device data sheet.</li> </ol><br>This field applies to the configuration of Port0 and Port1.<br>0000 No additional wait-states are added<br>0001 One additional wait-state is added<br>...<br>1111 Fifteen additional wait-states are added                                                                                                                                                                                                                                             |
| 24<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 25<br>P0_DPFEN | Port0 Data Prefetch Enable. This field enables or disables prefetching initiated by a data read access.<br><br>0 No prefetching is triggered by a data read access<br>1 Prefetching may be triggered by any data read access                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 26<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

*Table continues on the next page...*

**PFLASH\_PFCR1 field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 27<br>P0_IPFEN    | Port0 Instruction Prefetch Enable. This bit enables or disables prefetching initiated by an instruction read access.<br><br>0 No prefetching is triggered by an instruction read access<br>1 Prefetching may be triggered by any instruction read access                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 28<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 29–30<br>P0_PFLIM | Port0 PFlash Prefetch Limit. This field controls the prefetch algorithm used by the prefetch controller. This field defines a limit on the maximum number of sequential prefetches which will be attempted between buffer misses. In all situations when enabled, only a single prefetch is initiated on each buffer miss or hit.<br><br>00 No prefetching or buffering is performed.<br>01 The referenced line is prefetched on a buffer miss, that is, prefetch on miss.<br>10 The referenced line is prefetched on a buffer miss, or the next sequential line is prefetched on a buffer hit (if not already present), that is, prefetch on miss or hit.<br>11 The referenced line is prefetched on a buffer miss, or the next sequential line is prefetched on a buffer hit (if not already present), that is, prefetch on miss or hit. |
| 31<br>P0_BFEN     | Port0 PFlash Line Read Buffers Enable. This bit enables or disables line read buffer hits. It is also used to invalidate the buffers. This bit can only be updated while PFCR3[BFEN_LK]=0.<br><br>0 The line read buffers are disabled from satisfying read requests, and all buffer valid bits are cleared.<br>1 The line read buffers are enabled to satisfy read requests on hits. Buffer valid bits may be set when the buffers are successfully filled.                                                                                                                                                                                                                                                                                                                                                                               |

**25.5.2 Platform Flash Configuration Register 2 (PFLASH\_PFCR2)**

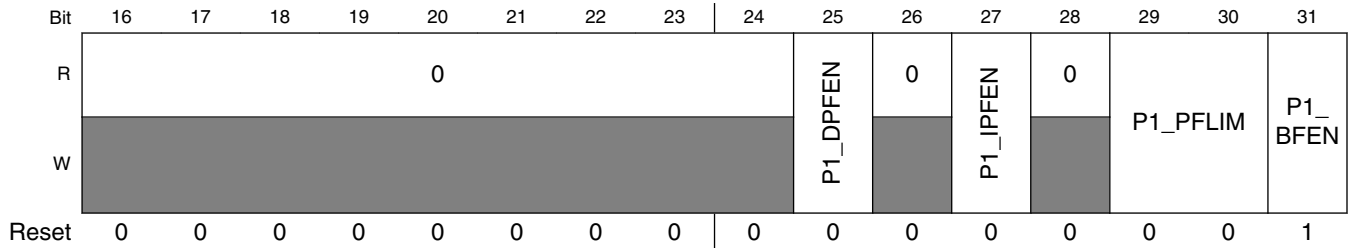
The PFlash Configuration Register 2 (PFCR2) controls the operation of Port1 of the PFLASH\_C55FM flash memory controller.

**NOTE**

See the chip-specific platform flash controller information for details about the masters.

Address: 0h base + 4h offset = 4h

| Bit   | 0         | 1         | 2         | 3         | 4         | 5         | 6        | 7        | 8        | 9        | 10       | 11       | 12       | 13       | 14       | 15       |
|-------|-----------|-----------|-----------|-----------|-----------|-----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| R     | P1_M15PFE | P1_M14PFE | P1_M13PFE | P1_M12PFE | P1_M11PFE | P1_M10PFE | P1_M9PFE | P1_M8PFE | P1_M7PFE | P1_M6PFE | P1_M5PFE | P1_M4PFE | P1_M3PFE | P1_M2PFE | P1_M1PFE | P1_M0PFE |
| W     |           |           |           |           |           |           |          |          |          |          |          |          |          |          |          |          |
| Reset | 0         | 0         | 0         | 0         | 0         | 0         | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        |

**PFLASH\_PFCR2 field descriptions**

| Field          | Description                                                                                                                                                                                                        |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>P1_M15PFE | Port1 Master 15 Prefetch enable. This bit controls whether prefetching may be triggered by AHB master 15.<br><br>0 No prefetching may be triggered by this master<br>1 Prefetching may be triggered by this master |
| 1<br>P1_M14PFE | Port1 Master 14 Prefetch enable. This bit controls whether prefetching may be triggered by AHB master 14.<br><br>0 No prefetching may be triggered by this master<br>1 Prefetching may be triggered by this master |
| 2<br>P1_M13PFE | Port1 Master 13 Prefetch enable. This bit controls whether prefetching may be triggered by AHB master 13.<br><br>0 No prefetching may be triggered by this master<br>1 Prefetching may be triggered by this master |
| 3<br>P1_M12PFE | Port1 Master 12 Prefetch enable. This bit controls whether prefetching may be triggered by AHB master 12.<br><br>0 No prefetching may be triggered by this master<br>1 Prefetching may be triggered by this master |
| 4<br>P1_M11PFE | Port1 Master 11 Prefetch enable. This bit controls whether prefetching may be triggered by AHB master 11.<br><br>0 No prefetching may be triggered by this master<br>1 Prefetching may be triggered by this master |
| 5<br>P1_M10PFE | Port1 Master 10 Prefetch enable. This bit controls whether prefetching may be triggered by AHB master 10.<br><br>0 No prefetching may be triggered by this master<br>1 Prefetching may be triggered by this master |
| 6<br>P1_M9PFE  | Port1 Master 9 Prefetch enable. This bit controls whether prefetching may be triggered by AHB master 9.<br><br>0 No prefetching may be triggered by this master<br>1 Prefetching may be triggered by this master   |
| 7<br>P1_M8PFE  | Port1 Master 8 Prefetch enable. This bit controls whether prefetching may be triggered by AHB master 8.<br><br>0 No prefetching may be triggered by this master<br>1 Prefetching may be triggered by this master   |
| 8<br>P1_M7PFE  | Port1 Master 7 Prefetch enable. This bit controls whether prefetching may be triggered by AHB master 7.                                                                                                            |

*Table continues on the next page...*

**PFLASH\_PFCR2 field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                       |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | 0 No prefetching may be triggered by this master<br>1 Prefetching may be triggered by this master                                                                                                                                                                                                                                 |
| 9<br>P1_M6PFE     | Port1 Master 6 Prefetch enable. This bit controls whether prefetching may be triggered by AHB master 6.<br>0 No prefetching may be triggered by this master<br>1 Prefetching may be triggered by this master                                                                                                                      |
| 10<br>P1_M5PFE    | Port1 Master 5 Prefetch enable. This bit controls whether prefetching may be triggered by AHB master 5.<br>0 No prefetching may be triggered by this master<br>1 Prefetching may be triggered by this master                                                                                                                      |
| 11<br>P1_M4PFE    | Port1 Master 4 Prefetch enable. This bit controls whether prefetching may be triggered by AHB master 4.<br>0 No prefetching may be triggered by this master<br>1 Prefetching may be triggered by this master                                                                                                                      |
| 12<br>P1_M3PFE    | Port1 Master 3 Prefetch enable. This bit controls whether prefetching may be triggered by AHB master 3.<br>0 No prefetching may be triggered by this master<br>1 Prefetching may be triggered by this master                                                                                                                      |
| 13<br>P1_M2PFE    | Port1 Master 2 Prefetch enable. This bit controls whether prefetching may be triggered by AHB master 2.<br>0 No prefetching may be triggered by this master<br>1 Prefetching may be triggered by this master                                                                                                                      |
| 14<br>P1_M1PFE    | Port1 Master 1 Prefetch enable. This bit controls whether prefetching may be triggered by AHB master 1.<br>0 No prefetching may be triggered by this master<br>1 Prefetching may be triggered by this master                                                                                                                      |
| 15<br>P1_M0PFE    | Port1 Master 0 Prefetch enable. This bit controls whether prefetching may be triggered by AHB master 0.<br>0 No prefetching may be triggered by this master<br>1 Prefetching may be triggered by this master                                                                                                                      |
| 16–24<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                           |
| 25<br>P1_DPFEN    | Port1 Data Prefetch Enable. This field enables or disables prefetching initiated by a data read access.<br>0 No prefetching is triggered by a data read access<br>1 Prefetching may be triggered by any data read access                                                                                                          |
| 26<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                           |
| 27<br>P1_IPFEN    | Port1 Instruction Prefetch Enable. This bit enables or disables prefetching initiated by an instruction read access.<br>0 No prefetching is triggered by an instruction read access<br>1 Prefetching may be triggered by any instruction read access                                                                              |
| 28<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                           |
| 29–30<br>P1_PFLIM | Port1 PFlash Prefetch Limit. This field controls the prefetch algorithm used by the prefetch controller. This field defines a limit on the maximum number of sequential prefetches which will be attempted between buffer misses. In all situations when enabled, only a single prefetch is initiated on each buffer miss or hit. |

*Table continues on the next page...*

**PFLASH\_PFCR2 field descriptions (continued)**

| Field         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               | 00 No prefetching or buffering is performed.<br>01 The referenced line is prefetched on a buffer miss, that is, prefetch on miss.<br>10 The referenced line is prefetched on a buffer miss, or the next sequential line is prefetched on a buffer hit (if not already present), that is, prefetch on miss or hit.<br>11 The referenced line is prefetched on a buffer miss, or the next sequential line is prefetched on a buffer hit (if not already present), that is, prefetch on miss or hit |
| 31<br>P1_BFEN | Port1 PFlash Line Read Buffers Enable. This bit enables or disables line read buffer hits. It is also used to invalidate the buffers. This bit can only be updated while PFCR3[BFEN_LK]=0.<br><br>0 The line read buffers are disabled from satisfying read requests, and all buffer valid bits are cleared.<br>1 The line read buffers are enabled to satisfy read requests on hits. Buffer valid bits may be set when the buffers are successfully filled.                                     |

**25.5.3 Platform Flash Configuration Register 3 (PFLASH\_PFCR3)**

Address: 0h base + 8h offset = 8h

|       |         |    |         |    |    |    |    |    |    |    |    |          |    |    |    |          |
|-------|---------|----|---------|----|----|----|----|----|----|----|----|----------|----|----|----|----------|
| Bit   | 0       | 1  | 2       | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11       | 12 | 13 | 14 | 15       |
| R     |         |    |         |    | 0  |    |    |    | 0  |    |    | Reserved |    | 0  |    |          |
| W     | P0_WCFG |    | P1_WCFG |    |    |    |    |    |    |    |    |          |    |    |    | BFEN_LK  |
| Reset | 0       | 0  | 0       | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0        | 0  | 0  | 0  | 0        |
| Bit   | 16      | 17 | 18      | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27       | 28 | 29 | 30 | 31       |
| R     |         |    | 0       |    |    |    |    |    | 0  |    |    |          |    |    |    |          |
| W     | ARBM    |    |         |    |    |    |    |    |    |    |    |          |    |    |    | DERR_SUP |
| Reset | 0       | 0  | 0       | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0        | 0  | 0  | 0  | 0        |

**PFLASH\_PFCR3 field descriptions**

| Field          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–1<br>P0_WCFG | Port0 Way Configuration. This field controls the configuration of the line buffers for a given set across all four ways in the controller cache. The indexed set can be organized as a "pool" of available resources, or with a fixed partition between instruction and data buffers.<br><br>In all cases, when a buffer miss occurs, the flash page is assigned a location within the controller cache. Within the indexed set, the way is selected using a least-recently-used replacement policy, and the entry is then marked as most-recently-used for that set. If the flash access is for the next-sequential line (prefetch), the buffer is not marked as most-recently-used until the given address produces a buffer hit.<br><br>This field is initialized by hardware reset.<br><br>00 All four buffers in an indexed set are available for any flash access, that is, there is no partitioning of the buffers based on the access type.<br>01 Reserved |

*Table continues on the next page...*

**PFLASH\_PFCR3 field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | <p>10 The buffers in an indexed set are partitioned into two groups with way 0 and 1 allocated for instruction fetches and way 2 and 3 for data accesses.</p> <p>11 The buffers are partitioned into two groups with way 0, 1, 2 allocated for instruction fetches and way 3 for data accesses.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 2–3<br>P1_WCFG    | <p>Port1 Way Configuration. This field controls the configuration of the line buffers for a given set across all four ways in the controller cache. The indexed set can be organized as a "pool" of available resources, or with a fixed partition between instruction and data buffers.</p> <p>In all cases, when a buffer miss occurs, the flash page is assigned a location within the controller cache. Within the indexed set, the way is selected using a least-recently-used replacement policy, and the entry is then marked as most-recently-used for that set. If the flash access is for the next-sequential line (prefetch), the buffer is not marked as most-recently-used until the given address produces a buffer hit.</p> <p>This field is initialized by hardware reset.</p> <p>00 All four buffers are available for any flash access, that is, there is no partitioning of the buffers based on the access type.</p> <p>01 Reserved</p> <p>10 The buffers are partitioned into two groups with ways 0 and 1 allocated for instruction fetches and ways 2 and 3 for data accesses.</p> <p>11 The buffers are partitioned into two groups with buffers 0,1,2 allocated for instruction fetches and buffer 3 for data accesses.</p> |
| 4–5<br>Reserved   | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 6–10<br>Reserved  | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 11<br>Reserved    | <p>This field is reserved.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 12–14<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 15<br>BFEN_LK     | <p>BFEN Lock. This field controls access to the PFCRm[Pn_BFEN] fields, which enable or disable line read buffer hits and are also used to invalidate the buffers.</p> <p>After this field is set to 1, it can be cleared to 0 only by a system reset.</p> <p>0 PFCRm[Pn_BFEN] fields are R/W.</p> <p>1 PFCRm[Pn_BFEN] fields are Read-only.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 16–17<br>ARBM     | <p>Arbitration Mode. This 2-bit field controls the arbitration of concurrent flash access requests from the two AHB ports of the flash memory controller. In both fixed priority or round-robin modes, write requests are prioritized higher than read requests, and read requests are prioritized higher than speculative prefetch requests whenever both ports issue concurrent requests.</p> <p>This field is initialized by hardware reset.</p> <p>00 Fixed priority arbitration with AHB p0&gt;p1</p> <p>01 Fixed priority arbitration with AHB p1&gt;p0</p> <p>10 Round-robin arbitration</p> <p>11 Round-robin arbitration</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 18–19<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 20–30<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

*Table continues on the next page...*



**PFLASH\_PFCR3 field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                                                                                                                                                                                           |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31<br>DERR_SUP | Data Error Suppression. Refer to the Embedded Flash Memory configuration information for which Flash blocks are affected by this field.<br><br>0 ECC events on data flash accesses are reported.<br>1 ECC events on data flash accesses are suppressed. Single-bit and multi-bit ECC errors are not reported. On multibit ECC events, the corrupted read data is replaced with a fixed, ECC-clean illegal opcode value of 0x15551555. |

**25.5.4 Platform Flash Access Protection Register (PFLASH\_PFAPR)**

The PFlash Access Protection Register (PFAPR) is used to control read and write accesses to the flash array.

**NOTE**

See the chip-specific platform flash controller information for details about the actual masters available on the device.

Address: 0h base + Ch offset = Ch

|       |      |      |      |      |      |      |      |      |   |   |    |    |    |    |    |    |
|-------|------|------|------|------|------|------|------|------|---|---|----|----|----|----|----|----|
| Bit   | 0    | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     |      |      |      |      |      |      |      |      |   |   |    |    |    |    |    |    |
| W     |      |      |      |      |      |      |      |      |   |   |    |    |    |    |    |    |
|       | M0AP | M1AP | M2AP | M3AP | M4AP | M5AP | M6AP | M7AP |   |   |    |    |    |    |    |    |
| Reset | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1    | 1 | 1 | 1  | 1  | 1  | 1  | 1  | 1  |

|       |      |      |       |       |       |       |       |       |    |    |    |    |    |    |    |    |
|-------|------|------|-------|-------|-------|-------|-------|-------|----|----|----|----|----|----|----|----|
| Bit   | 16   | 17   | 18    | 19    | 20    | 21    | 22    | 23    | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |      |      |       |       |       |       |       |       |    |    |    |    |    |    |    |    |
| W     |      |      |       |       |       |       |       |       |    |    |    |    |    |    |    |    |
|       | M8AP | M9AP | M10AP | M11AP | M12AP | M13AP | M14AP | M15AP |    |    |    |    |    |    |    |    |
| Reset | 1    | 1    | 1     | 1     | 1     | 1     | 1     | 1     | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  |

**PFLASH\_PFAPR field descriptions**

| Field       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–1<br>M0AP | Master 0 Access Protection. This field controls whether read and write accesses to the flash are allowed based on the master ID of a requesting master. These fields are initialized by hardware reset.<br><br>00 No accesses may be performed by this master<br>01 Only read accesses may be performed by this master<br>10 Only write accesses may be performed by this master<br>11 Both read and write accesses may be performed by this master |
| 2–3<br>M1AP | Master 1 Access Protection. This field controls whether read and write accesses to the flash are allowed based on the master ID of a requesting master. These fields are initialized by hardware reset.<br><br>00 No accesses may be performed by this master<br>01 Only read accesses may be performed by this master<br>10 Only write accesses may be performed by this master<br>11 Both read and write accesses may be performed by this master |
| 4–5<br>M2AP | Master 2 Access Protection. This field controls whether read and write accesses to the flash are allowed based on the master ID of a requesting master. These fields are initialized by hardware reset.                                                                                                                                                                                                                                             |

*Table continues on the next page...*

**PFLASH\_PFAPR field descriptions (continued)**

| Field         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               | 00 No accesses may be performed by this master<br>01 Only read accesses may be performed by this master<br>10 Only write accesses may be performed by this master<br>11 Both read and write accesses may be performed by this master                                                                                                                                                                                                                |
| 6–7<br>M3AP   | Master 3 Access Protection. This field controls whether read and write accesses to the flash are allowed based on the master ID of a requesting master. These fields are initialized by hardware reset.<br><br>00 No accesses may be performed by this master<br>01 Only read accesses may be performed by this master<br>10 Only write accesses may be performed by this master<br>11 Both read and write accesses may be performed by this master |
| 8–9<br>M4AP   | Master 4 Access Protection. This field controls whether read and write accesses to the flash are allowed based on the master ID of a requesting master. These fields are initialized by hardware reset.<br><br>00 No accesses may be performed by this master<br>01 Only read accesses may be performed by this master<br>10 Only write accesses may be performed by this master<br>11 Both read and write accesses may be performed by this master |
| 10–11<br>M5AP | Master 5 Access Protection. This field controls whether read and write accesses to the flash are allowed based on the master ID of a requesting master. These fields are initialized by hardware reset.<br><br>00 No accesses may be performed by this master<br>01 Only read accesses may be performed by this master<br>10 Only write accesses may be performed by this master<br>11 Both read and write accesses may be performed by this master |
| 12–13<br>M6AP | Master 6 Access Protection. This field controls whether read and write accesses to the flash are allowed based on the master ID of a requesting master. These fields are initialized by hardware reset.<br><br>00 No accesses may be performed by this master<br>01 Only read accesses may be performed by this master<br>10 Only write accesses may be performed by this master<br>11 Both read and write accesses may be performed by this master |
| 14–15<br>M7AP | Master 7 Access Protection. This field controls whether read and write accesses to the flash are allowed based on the master ID of a requesting master. These fields are initialized by hardware reset.<br><br>00 No accesses may be performed by this master<br>01 Only read accesses may be performed by this master<br>10 Only write accesses may be performed by this master<br>11 Both read and write accesses may be performed by this master |
| 16–17<br>M8AP | Master 8 Access Protection. This field controls whether read and write accesses to the flash are allowed based on the master ID of a requesting master. These fields are initialized by hardware reset.<br><br>00 No accesses may be performed by this master<br>01 Only read accesses may be performed by this master<br>10 Only write accesses may be performed by this master<br>11 Both read and write accesses may be performed by this master |
| 18–19<br>M9AP | Master 9 Access Protection. This field controls whether read and write accesses to the flash are allowed based on the master ID of a requesting master. These fields are initialized by hardware reset.                                                                                                                                                                                                                                             |

*Table continues on the next page...*

**PFLASH\_PFAPR field descriptions (continued)**

| <b>Field</b>   | <b>Description</b>                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | 00 No accesses may be performed by this master<br>01 Only read accesses may be performed by this master<br>10 Only write accesses may be performed by this master<br>11 Both read and write accesses may be performed by this master                                                                                                                                                                                                                 |
| 20–21<br>M10AP | Master 10 Access Protection. This field controls whether read and write accesses to the flash are allowed based on the master ID of a requesting master. These fields are initialized by hardware reset.<br><br>00 No accesses may be performed by this master<br>01 Only read accesses may be performed by this master<br>10 Only write accesses may be performed by this master<br>11 Both read and write accesses may be performed by this master |
| 22–23<br>M11AP | Master 11 Access Protection. This field controls whether read and write accesses to the flash are allowed based on the master ID of a requesting master. These fields are initialized by hardware reset.<br><br>00 No accesses may be performed by this master<br>01 Only read accesses may be performed by this master<br>10 Only write accesses may be performed by this master<br>11 Both read and write accesses may be performed by this master |
| 24–25<br>M12AP | Master 12 Access Protection. This field controls whether read and write accesses to the flash are allowed based on the master ID of a requesting master. These fields are initialized by hardware reset.<br><br>00 No accesses may be performed by this master<br>01 Only read accesses may be performed by this master<br>10 Only write accesses may be performed by this master<br>11 Both read and write accesses may be performed by this master |
| 26–27<br>M13AP | Master 13 Access Protection. This field controls whether read and write accesses to the flash are allowed based on the master ID of a requesting master. These fields are initialized by hardware reset.<br><br>00 No accesses may be performed by this master<br>01 Only read accesses may be performed by this master<br>10 Only write accesses may be performed by this master<br>11 Both read and write accesses may be performed by this master |
| 28–29<br>M14AP | Master 14 Access Protection. This field controls whether read and write accesses to the flash are allowed based on the master ID of a requesting master. These fields are initialized by hardware reset.<br><br>00 No accesses may be performed by this master<br>01 Only read accesses may be performed by this master<br>10 Only write accesses may be performed by this master<br>11 Both read and write accesses may be performed by this master |
| 30–31<br>M15AP | Master 15 Access Protection. This field controls whether read and write accesses to the flash are allowed based on the master ID of a requesting master. These fields are initialized by hardware reset.<br><br>00 No accesses may be performed by this master<br>01 Only read accesses may be performed by this master<br>10 Only write accesses may be performed by this master<br>11 Both read and write accesses may be performed by this master |

## 25.6 Functional description

The flash memory controller interfaces between:

- The AHB system bus ports
- Flash memory array

For accesses targeting the flash array, the flash memory controller generates read and write enables, block selects, array address, write size and write data as inputs to the flash array. The flash memory controller captures read data from the flash array and drives it onto the AHB system bus. Up to four pages of data (256-bit page size) may be buffered in each of four ways of the flash memory controller mini-cache. Lines may be prefetched in advance of being requested, allowing single-cycle (zero AHB wait-states) read data responses on buffer hits.

Several prefetch control algorithms are available for controlling line read buffer fills. Prefetch triggering may be restricted to instruction accesses only, data accesses only, or may be unrestricted. Prefetch triggering may also be controlled on a per-master basis.

Buffers may also be selectively enabled or disabled for allocation by instruction and data prefetch.

Access protections may be applied on a per-master basis for both reads and writes to support security and privilege mechanisms.

### 25.6.1 Basic interface protocol

Read accesses are terminated under control of the appropriate wait state settings. Thus, the access time of the operation is determined by the setting of the PFCR1[RWSC] field. Access timing can be varied to account for the operating conditions of the SoC (frequency, voltage, temp) by appropriately setting the PFCR1[RWSC] field.

### 25.6.2 Access protections

On-chip Flash memory accesses are subject to restrictions based on chip security implementation and/or requirements based on functional restrictions of the flash memory, e.g., a read access to a flash block while it is currently being written is not allowed. When a prohibited access is attempted, a system bus error termination results.

The flash memory controller may invoke a system bus error termination in the following scenarios:

- Attempted access by an AHB master whose corresponding read access control or write access control settings do not allow the access, thus causing a protection violation. See [Platform Flash Access Protection Register \(PFLASH\\_PFAPR\)](#) for more detail. In this case the flash memory controller does not initiate the requested flash array access.
- The flash returns an error response on an attempted access to a partition that is unavailable. (see [Embedded Flash Memory](#)).
- Attempted access by an AHB master to a reserved region in the flash memory map.

The flash array may also signal an error response to terminate a requested access due to improper sequencing during program/erase operations, improper sequencing during array integrity testing. When an error response is received the flash memory controller does not update or validate a page read buffer. An error response may be signaled on a read or interlock write operation. For more information on the specifics related to signaling of flash errors, including flash ECC events, array integrity testing and read-while-write events, refer to the flash memory chapter.

### 25.6.2.1 PFAPR - Platform Flash Access Protection Register (PFAPR)

The flash memory controller provides programmable, configurable access protections for both read and write cycles on a per-master basis via the PFlash Access Protection Register (PFAPR). It allows restriction of read and write requests on a per-master basis. This functionality is described in [Platform Flash Access Protection Register \(PFLASH\\_PFAPR\)](#). Detection of a protection violation results in an error response from the flash memory controller on the AHB transfer.

### 25.6.3 Access pipelining

Accesses to the flash array may be pipelined by driving a subsequent access address and control signals while waiting for the current access to complete. Pipelined access requests are always run to completion and are not aborted by the flash memory controller. Flash access pipelining allows for improved performance by reducing the access latency seen by the AHB master. Access pipelining may be applied only to read cycles targeting the flash array. Address pipelining is enabled by setting the PFCR1[APC] field.

Access pipelining is only supported for normal flash access .

#### NOTE

Not all combinations of PFCR1[APC] (Address Pipeline Control) and PFCR1[RWSC] (Read Wait State Control)

settings are recommended or supported. Please refer to the device data sheet for guidance.

## 25.6.4 Line read buffers and prefetch operation

The AHB ports of the flash memory controller each contain a four-way set-associative mini cache, where each way contains four page buffers which are used to hold data read from the flash arrays. Each 256-bit buffer operates independently, and is filled using a single array access. The buffers are used for both prefetch and normal demand fetches.

Prefetch triggering is controllable on a per-master and access-type basis (see [Platform Flash Configuration Register 1 \(PFLASH\\_PFCR1\)](#)). Bus masters may be enabled or disabled from triggering prefetches, and triggering may be further restricted based on whether a read access is for instruction or data. A read access by the flash memory controller may trigger a prefetch to the next sequential line of array data on the cycle following the request. The access address is incremented by 32-bytes, and a subsequent flash access is initiated. A flash array prefetch is initiated if the data is not already resident in a line read buffer. Prefetched data is always loaded into the least-recently-used buffer.

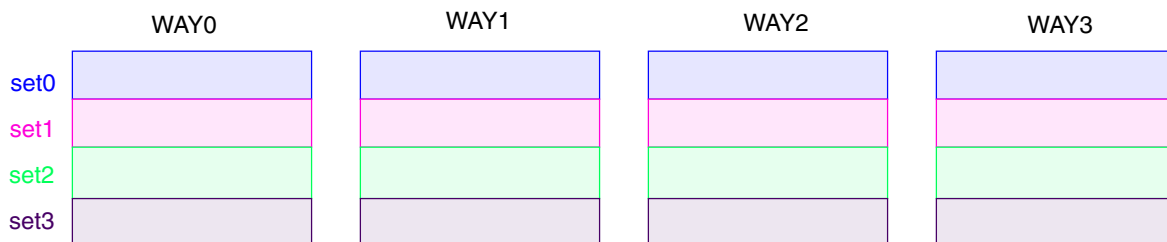


**Figure 25-6. Flash memory controller 4-entry, 4-way mini-cache organization**

Once the candidate line buffer has been selected, the flash array is accessed and read data loaded into the buffer. If the buffer load was in response to a miss, the buffer which was loaded is immediately marked as most-recently-used. If the buffer load was in response to a speculative fetch to the next-sequential line address after a buffer hit, *the recently-used status is not changed*. Rather, it is marked as most-recently-used only after a subsequent buffer hit.

This policy maximizes performance based on reference patterns of flash accesses and allows for prefetched data to remain valid when non-prefetch enabled bus masters are granted flash access.

Several algorithms are available for prefetch control which trade off performance for power. They are described in [Platform Flash Configuration Register 1 \(PFLASH\\_PFCR1\)](#). More aggressive prefetching may increase power due to the number of potentially discarded prefetches, but may increase performance by lowering average read latency.

For prefetching to occur, all of the following must apply:

1. PFCR{1,2}[P{0,1}\_BFEN] must be set to '1',
2. PFCR{1,2}[P{0,1}\_PFLIM] must be non-zero, and
3. Either PFCR{1,2}[P{0,1}\_IPFEN] or PFCR{1,2}[P{0,1}\_DPFEN] must be asserted.

Refer to [Platform Flash Configuration Register 1 \(PFLASH\\_PFCR1\)](#) for a description of these controls.

### 25.6.5 Instruction/Data prefetch triggering

Port0 prefetch triggering may be enabled for instruction reads via the PFCR1[IPFEN] control field, while Port0 prefetching for data reads is enabled via the PFCR1[DPFEN] control field. Additionally, the PFCR1[PFLIM] must also be set to enable prefetching on Port0. Refer to [Platform Flash Configuration Register 1 \(PFLASH\\_PFCR1\)](#) for a description of these controls. Prefetches are never triggered by write cycles.

Port1 prefetch triggering may be enabled for instruction reads via the PFCR2[IPFEN] control field, while Port1 prefetching for data reads is enabled via the PFCR2[DPFEN] control field. Additionally, the PFCR2[PFLIM] must also be set to enable prefetching on Port1. Refer to [Platform Flash Configuration Register 2 \(PFLASH\\_PFCR2\)](#) for a description of these controls. Prefetches are never triggered by write cycles.

### 25.6.6 Per-Master prefetch triggering

Prefetch triggering may be controlled for individual bus masters. Refer to [Platform Flash Configuration Register 1 \(PFLASH\\_PFCR1\)](#) for a description of these controls.

### 25.6.7 Buffer allocation

Allocation of the line read buffers is controlled via the PFCR3 control register, specifically the line buffer configuration ({P0,P1}\_WCFG) field. Refer to [Platform Flash Configuration Register 3 \(PFLASH\\_PFCR3\)](#). The buffers can be organized as a "pool" of available resources (with all four ways within a given set) or with a fixed partition between ways allocated to instruction or data accesses. For the fixed partitions, two

configurations are supported. In one configuration, ways 0 and 1 are allocated for instruction fetches and ways 2 and 3 for data accesses. In the second configuration, ways 0, 1 and 2 are allocated for instruction fetches and way 3 reserved for data accesses.

## 25.6.8 Reliability considerations

This section summarizes mechanism to enhance the reliability of flash by correcting and detecting faults.

### 25.6.8.1 e2eECC End-to-End ECC

#### 25.6.8.1.1 e2eECC and flash accesses

There are multiple complications associated with the use of the standard e2eECC algorithm with flash memory. The basic issues involve: the default state of a flash block that is erased (all ones) and the fact that programming an erased flash involves changing the state of memory bit from a logical 1 to a logical 0. These factors require that the system level e2eECC must be modified on references to the flash memory in three ways.

First, the all-ones codeword, since it reflects the state of an erased flash location, is treated as a valid error free encoding; in particular, it is required that the checkbits associated with the all all-ones value and the all zeroes data value are required to be the same with a checkbit value of 0xFF. Second, since an erased flash block generates an all-ones data for all locations, the address field cannot be included in the ECC code used by the flash. The flash array implements special address re-encoding logic based on the decoded array address to protect against address faults. Third, there are additional ECC implications associated with the flash memory and its support for EEPROM emulation.

In summary, the basic operation of an NVM bit cell impacts the e2eECC algorithm used in the flash memory in multiple ways. The required ECC adjustments are handled by the platform flash controller before data is written to the flash array(s) or applied to read data accessed from the array(s). These adjustments are two-fold. Consider a flash write during a programming event:

1. Remove the address field from the ECC codeword by XOR'ing the address calculation of the H-matrix from the checkbits.
2. Invert the resulting 8-bit ECC checkbit vector. This step is required to support the all-ones erased state.

On a flash read operation, a similar but "reversed" set of steps are performed as the data is driven into the system bus interconnect:



1. Invert the 8-bit ECC checkbit vector read from the flash.
2. Factor the address field into the ECC checkbits by XOR'ing the address calculation of the H-matrix.

Consider the following flash ECC example. Let the flash block containing address 0x00345678 be erased. The following is the sequence of operations:

1. Read address 0x00345678. Flash array returns fl\_rdata = 0xFFFFFFFF\_FFFFFFFF, fl\_cdata = 0xFF. The flash memory controller calculates the addr\_chkbit = 0xC4 using the H-matrix; it inverts the fl\_cdata vector to 0x00 and factors in the addr\_chkbit = 0xC4 to produce the following valid e2eECC codeword:

addr = 0x00345678, rdata = 0xFFFFFFFF\_FFFFFFFF, rchkbit = 0xC4 ( $0x00 \wedge 0xC4$ )

2. Program address 0x00345678 with data = 0xBABEFACE\_DEADBEEF. The initiating bus master calculates the e2eECC codeword as:

addr = 0x00345678, wdata = 0xBABEFACE\_DEADBEEF, wchkbit = 0xE3

Before performing the interlock write to the flash array, the flash memory controller adjusts the chkbit value as fl\_wchkbit = 0xD8 ( $0xE3 \wedge 0xFF \wedge 0xC4$ ) by toggling the original write checkbits and then factoring in the addr\_ecc. Accordingly, the codeword stored in the flash array is:

fl\_data = 0xBABEFACE\_DEADBEEF, fl\_checkbit = 0xD8

3. Read address 0x00345678. The flash array returns the stored codeword, fl\_rdata = 0xBABEFACE\_DEADBEEF, fl\_rchkbit = 0xD8. The flash memory controller inverts the checkbit vector and factors in the address checkbits ( $0xD8 \wedge 0xFF \wedge 0xC4$ ) to produce:

addr = 0x00345678, rdata = 0xBABEFACE\_DEADBEEF, rchkbit = 0xE3

This read codeword matches the original e2eECC codeword generated by the write.

Malfunction of the ECC logic described above may result in the corruption of the SEC/DED event reporting. The flash memory controller performs EDC after ECC check on all flash transactions. If a mismatch is detected, indicating a failure in the ECC logic, the event is reported to the device fault collection module.

### 25.6.8.2 Flash address generation check

Fault detection coverage of the address and data are handled by ECC performed within the flash and e2eECC performed at the master. Fault detection coverage of the address path and control within the flash memory controller rely on a feedback path between the flash controller and flash. Recall on a requested access to flash, the flash memory controller must decode the system AHB bus signals to generate the corresponding flash interface signals to invoke a flash lookup. In addition to providing the requested read data, the flash also provides output sidebands reflecting the encoded address and block selects used to perform the actual row lookup.

This sideband information is used by the flash memory controller to verify the expected transaction. If a mismatch is detected, indicating a failure in the address generation or control logic within the flash memory controller or the transmission path between the flash memory controller and flash array, the event is forwarded to the device fault collection module and the corresponding buffer is invalidated.

### 25.6.9 ECC on data flash accesses

For single-bit correction events, the corrected data and checkbits are returned to the requesting master.

In the event of a non-correctable error detection, a fixed, illegal opcode value (1555\_1555h) is returned to the requesting master along with the associated ECC checkbits as determined by the requesting address.

When PFCR3[DERR\_SUP] is enabled, ECC errors on data flash blocks are suppressed from being reported.

- In the event of a single-bit correction, the corrected data and checkbits are returned to the requesting master, and the single-bit correction event is suppressed from being reported.
- In the event of a non-correctable error detection, a fixed, illegal opcode value (1555\_1555h) is returned to the requesting master along with the associated ECC checkbits as determined by the requesting address and the non-correctable error event is suppressed from being reported.

#### NOTE

EEPROM should be avoided for storage of executable code.

### 25.6.10 Array integrity considerations

During an array integrity sequence, the flash memory array ignores any incoming read requests. When a flash array integrity check is in progress, the flash memory controller terminates all flash access requests with an error. More specifically, it aborts the incoming flash access requests and terminates the system bus transfer with an error.



# Chapter 26

## Embedded Flash Memory (C55FMC)

### 26.1 Chip-specific C55FMC information

#### NOTE

In production devices of this chip, the one-time-programmable Test Mode Disable Seal is programmed—that is, UTest is sealed—at the factory. For more information about the seal, see [Test mode disable seal](#).

#### 26.1.1 C55FMC register reset values

The following table lists C55FMC register reset values that are specific to this chip. For the reset values of all other C55FMC registers, see [Memory map and register definition](#).

Table 26-1. C55FMC register reset values

| Register                                                             | Reset value                                 |
|----------------------------------------------------------------------|---------------------------------------------|
| <a href="#">Extended Module Configuration Register (C55FMC_MCRE)</a> | 1001_2020h                                  |
| <a href="#">Over-Program Protection 0 register (C55FMC_OPP0)</a>     | Dependent on customer settings <sup>1</sup> |
| <a href="#">Over-Program Protection 1 register (C55FMC_OPP1)</a>     |                                             |
| <a href="#">Over-Program Protection 2 register (C55FMC_OPP2)</a>     |                                             |
| <a href="#">Over-Program Protection 3 register (C55FMC_OPP3)</a>     |                                             |

1. The reset value of the C55FMC\_OPPn registers, for each flash memory block mapped to a bit within each register, reflects whether the block has been marked as One Time Programmable (OTP). The C55FMC\_OPPn registers have the same bit-to-block mapping as the C55FMC\_LOCKn registers. A value of 1 indicates a block has been marked as OTP. The reset values of the C55FMC\_OPPn are similar to the reset values of the TDM\_OTPENn registers in the Tamper Detection Module (TDM).

The only C55FMC register that configures the use of the CSE-exclusive blocks is the OPP1 register. The LOCK1 and SEL1 registers do not configure these blocks.

### 26.1.3 C55FMC\_LOCKn register bit mapping

### 26.1.3.1 C55FMC\_LOCK0 register bit mapping

[illegible]

### 26.1.3.2 C55FMC\_LOCK1 register bit mapping

The flash memory blocks mapped to the **Lock 1 register (C55FMC\_LOCK1)** bits are as follows.

## NOTE

This register cannot be used to manage the blocks mapped to the register's implemented bits because the CSE module has exclusive access to these blocks. The bits' value is 1 to reflect the corresponding blocks' locked state, and the bits cannot be written to change that state.

| Register                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| Flash Memory Block Name                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Register Field                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| <div> <div> <div>0</div> <div>1</div> <div>2</div> <div>3</div> <div>4</div> <div>5</div> <div>6</div> <div>7</div> <div>8</div> <div>9</div> <div>10</div> <div>11</div> <div>12</div> <div>13</div> <div>14</div> <div>15</div> <div>16</div> <div>17</div> <div>18</div> <div>19</div> <div>20</div> <div>21</div> <div>22</div> <div>23</div> <div>24</div> <div>25</div> <div>26</div> <div>27</div> <div>28</div> <div>29</div> <div>30</div> <div>31</div> </div> <div> <div>Reserved</div> <div>HIGHLOCK[15:0]</div> </div> </div> |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| <div> <div>—</div> <div>—</div> <div>—</div> <div>—</div> <div>—</div> <div>—</div> <div>—</div> <div>—</div> <div>—</div> <div>—</div> <div>—</div> <div>—</div> <div>—</div> <div>—</div> <div>—</div> <div>—</div> <div>—</div> <div>—</div> <div>—</div> <div>—</div> <div>—</div> <div>—</div> <div>—</div> <div>—</div> <div>—</div> <div>—</div> <div>—</div> <div>—</div> <div>—</div> <div>—</div> <div>—</div> </div> <div> <div>16 KB CSE Data Block1 - Partition 5</div> <div>16 KB CSE Data Block0 - Partition 4</div> </div> |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

### 26.1.3.3 C55FMC\_LOCK2 register bit mapping

The flash memory blocks mapped to the [Lock 2 register \(C55FMC\\_LOCK2\)](#) bits are as follows.

|   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|

*Table continues on the next page...*

| Flash Memory Block Name                       | Register Field  |
|-----------------------------------------------|-----------------|
| 256 KB Boot Code Block0                       | A256KLOCK[15:0] |
| 256 KB Application Code BlockX                |                 |
| 256 KB Application Code BlockX                |                 |
| 256 KB Application Code BlockX                |                 |
| 256 KB Application Code BlockX                |                 |
| 256 KB Application Code Block26 - Partition 9 |                 |
| 256 KB Application Code Block25 - Partition 9 |                 |
| 256 KB Application Code Block24 - Partition 9 |                 |
| 256 KB Application Code Block23 - Partition 8 |                 |
| 256 KB Application Code Block22 - Partition 8 |                 |
| 256 KB Application Code Block21 - Partition 8 |                 |
| 256 KB Application Code Block20 - Partition 8 |                 |
| 256 KB Application Code Block19 - Partition 8 |                 |
| 256 KB Application Code Block18 - Partition 8 |                 |
| 256 KB Application Code Block17 - Partition 8 |                 |
| 256 KB Application Code Block16 - Partition 8 |                 |
| Reserved                                      |                 |

26.1.3.4 C55FMC\_LOCK3 register bit mapping

The flash memory blocks mapped to the [Lock 3 register \(C55FMC\\_LOCK3\)](#) bits are as follows.

|                | 0        | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16              | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|----------------|----------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|-----------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Register Field | Reserved |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | A256KLOCK[15:0] |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

Table continues on the next page...



[illegible]

#### 26.1.4 C55FMC\_SEL $n$ register bit mapping

The C55FMC\_SEL*n* registers provide a means to select flash memory blocks to be erased. Each implemented bit in a C55FMC\_SEL*n* register maps to a specific memory block.

#### 26.1.4.1 C55FMC\_SEL0 register bit mapping

The flash memory blocks mapped to the [Select 0 register \(C55FMC\\_SEL0\)](#) bits are as follows.

|                |          | 0            | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13           | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|----------------|----------|--------------|---|---|---|---|---|---|---|---|---|----|----|----|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Register Field | Reserved | LOWSEL[13:0] |   |   |   |   |   |   |   |   |   |    |    |    | MIDSEL[15:0] |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|                |          |              |   |   |   |   |   |   |   |   |   |    |    |    |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

*Table continues on the next page...*

| Flash Memory Block Name                |
|----------------------------------------|
| —                                      |
| —                                      |
| —                                      |
| —                                      |
| —                                      |
| —                                      |
| —                                      |
| —                                      |
| —                                      |
| —                                      |
| —                                      |
| 64 KB EEPROM Data Block1 - Partition 1 |
| 64 KB EEPROM Data Block0 - Partition 0 |
| —                                      |
| —                                      |
| —                                      |
| —                                      |
| —                                      |
| —                                      |
| —                                      |
| —                                      |
| —                                      |
| —                                      |
| 64 KB EEPROM Data Block1 - Partition 3 |
| 64 KB EEPROM Data Block0 - Partition 2 |

#### 26.1.4.2 C55FMC\_SEL1 register bit mapping

The flash memory blocks mapped to the [Select 1 register \(C55FMC\\_SEL1\)](#) bits are as follows.

## NOTE

This register cannot be used to select the blocks mapped to the register's implemented bits because the CSE module has exclusive access to the blocks.

| 0              |          | 1 |  | 2 |  | 3 |  | 4 |  | 5 |  | 6 |  | 7 |  | 8 |  | 9 |  | 10 |  | 11 |  | 12 |  | 13 |  | 14 |  | 15 |               | 16 |  | 17 |  | 18 |  | 19 |  | 20 |  | 21 |  | 22 |  | 23 |  | 24 |  | 25 |  | 26 |  | 27 |  | 28 |  | 29 |  | 30 |  | 31 |  |
|----------------|----------|---|--|---|--|---|--|---|--|---|--|---|--|---|--|---|--|---|--|----|--|----|--|----|--|----|--|----|--|----|---------------|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|----|--|
| Register Field | Reserved |   |  |   |  |   |  |   |  |   |  |   |  |   |  |   |  |   |  |    |  |    |  |    |  |    |  |    |  |    | HIGHSEL[15:0] |    |  |    |  |    |  |    |  |    |  |    |  |    |  |    |  |    |  |    |  |    |  |    |  |    |  |    |  |    |  |    |  |

*Table continues on the next page...*

[illegible]

#### 26.1.4.3 C55FMC\_SEL2 register bit mapping

The flash memory blocks mapped to the [Select 2 register \(C55FMC\\_SEL2\)](#) bits are as follows.

| 0              | 1              | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16       | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |  |
|----------------|----------------|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--|
| Register Field | A256KSEL[15:0] |   |   |   |   |   |   |   |   |    |    |    |    |    |    | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |
|                |                |   |   |   |   |   |   |   |   |    |    |    |    |    |    |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |

*Table continues on the next page...*

| Flash Memory Block Name                       |          |
|-----------------------------------------------|----------|
| 256 KB Boot Code Block0                       |          |
| 256 KB Application Code BlockX                |          |
| 256 KB Application Code BlockX                |          |
| 256 KB Application Code BlockX                |          |
| 256 KB Application Code BlockX                |          |
| 256 KB Application Code Block26 - Partition 9 |          |
| 256 KB Application Code Block25 - Partition 9 |          |
| 256 KB Application Code Block24 - Partition 9 |          |
| 256 KB Application Code Block23 - Partition 8 |          |
| 256 KB Application Code Block22 - Partition 8 |          |
| 256 KB Application Code Block21 - Partition 8 |          |
| 256 KB Application Code Block20 - Partition 8 |          |
| 256 KB Application Code Block19 - Partition 8 |          |
| 256 KB Application Code Block18 - Partition 8 |          |
| 256 KB Application Code Block17 - Partition 8 |          |
| 256 KB Application Code Block16 - Partition 8 |          |
|                                               | Reserved |

#### 26.1.4.4 C55FMC\_SEL3 register bit mapping

The flash memory blocks mapped to the [Select 3 register \(C55FMC\\_SEL3\)](#) bits are as follows.

| 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 |          |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|---------------------------------------------------------------------------------------|----------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|----------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| Register Field                                                                        | Reserved |  |  |  |  |  |  |  |  |  |  |  |  |  |  | A256KSEL[15:0] |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Flash Memory Block Name                                                               |          |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

## 26.1.5 C55FMC\_OPP $n$ register bit mapping

The C55FMC\_OPP $n$  registers provide a means to protect blocks from being over-programmed. Each implemented bit in a C55FMC\_OPP $n$  register maps to a specific memory block.

### 26.1.5.1 C55FMC\_OPP0 register bit mapping

The flash memory blocks mapped to the [Over-Program Protection 0 register \(C55FMC\\_OPP0\)](#) bits are as follows.

|                                        | 0        | 1        | 2            | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17           | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|----------------------------------------|----------|----------|--------------|---|---|---|---|---|---|---|----|----|----|----|----|----|----|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Register Field                         | Reserved | Reserved | LOWOPP[13:0] |   |   |   |   |   |   |   |    |    |    |    |    |    |    | MIDOPP[15:0] |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Flash Memory Block Name                | —        | —        | —            | — | — | — | — | — | — | — | —  | —  | —  | —  | —  | —  | —  | —            | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  |    |
| 64 KB EEPROM Data Block1 - Partition 1 |          |          |              |   |   |   |   |   |   |   |    |    |    |    |    |    |    |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 64 KB EEPROM Data Block0 - Partition 0 |          |          |              |   |   |   |   |   |   |   |    |    |    |    |    |    |    |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|                                        | —        | —        | —            | — | — | — | — | — | — | — | —  | —  | —  | —  | —  | —  | —  | —            | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  |    |
| 64 KB EEPROM Data Block1 - Partition 3 |          |          |              |   |   |   |   |   |   |   |    |    |    |    |    |    |    |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 64 KB EEPROM Data Block0 - Partition 2 |          |          |              |   |   |   |   |   |   |   |    |    |    |    |    |    |    |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

### 26.1.5.2 C55FMC\_OPP1 register bit mapping

The flash memory blocks mapped to the [Over-Program Protection 1 register \(C55FMC\\_OPP1\)](#) bits are as follows.

#### NOTE

When the value of a bit in this register reflects that the corresponding flash block has over-program protection, the

Chip-specific C55FMC information

protection applies to the block even though the CSE module has exclusive access to the block.

|                         | 0        | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15            | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31                                  |                                     |
|-------------------------|----------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|---------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------------------------------|-------------------------------------|
| Register Field          | Reserved |   |   |   |   |   |   |   |   |   |    |    |    |    |    | HIGHOPP[15:0] |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                                     |                                     |
| Flash Memory Block Name | —        | — | — | — | — | — | — | — | — | — | —  | —  | —  | —  | —  | —             | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | 16 KB CSE Data Block1 - Partition 5 | 16 KB CSE Data Block0 - Partition 4 |

26.1.5.3 C55FMC\_OPP2 register bit mapping

The flash memory blocks mapped to the [Over-Program Protection 2 register \(C55FMC\\_OPP2\)](#) bits are as follows.

|                | 0              | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15                                   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|----------------|----------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|--------------------------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Register Field | A256KOPP[15:0] |   |   |   |   |   |   |   |   |   |    |    |    |    |    | R<br>e<br>s<br>e<br>r<br>v<br>e<br>d |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

Table continues on the next page...

| Flash Memory Block Name                       |
|-----------------------------------------------|
| 256 KB Boot Code Block0                       |
| 256 KB Application Code BlockX                |
| 256 KB Application Code BlockX                |
| 256 KB Application Code BlockX                |
| 256 KB Application Code BlockX                |
| 256 KB Application Code BlockX                |
| 256 KB Application Code Block26 - Partition 9 |
| 256 KB Application Code Block25 - Partition 9 |
| 256 KB Application Code Block24 - Partition 9 |
| 256 KB Application Code Block23 - Partition 8 |
| 256 KB Application Code Block22 - Partition 8 |
| 256 KB Application Code Block21 - Partition 8 |
| 256 KB Application Code Block20 - Partition 8 |
| 256 KB Application Code Block19 - Partition 8 |
| 256 KB Application Code Block18 - Partition 8 |
| 256 KB Application Code Block17 - Partition 8 |
| 256 KB Application Code Block16 - Partition 8 |
| Reserved                                      |

#### 26.1.5.4 C55FMC\_OPP3 register bit mapping

The flash memory blocks mapped to the [Over-Program Protection 3 register \(C55FMC\\_OPP3\)](#) bits are as follows.

|                         | 0        | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15             | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|-------------------------|----------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Register Field          | Reserved |   |   |   |   |   |   |   |   |   |    |    |    |    |    | A256KOPP[15:0] |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Flash Memory Block Name |          |   |   |   |   |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

## 26.2 Introduction

The primary function of the embedded flash memory is to serve as electrically programmable and erasable non-volatile memory (NVM) that may be used for instruction or data storage.

The following figure shows the top-level diagram and functional organization of the flash memory unit.

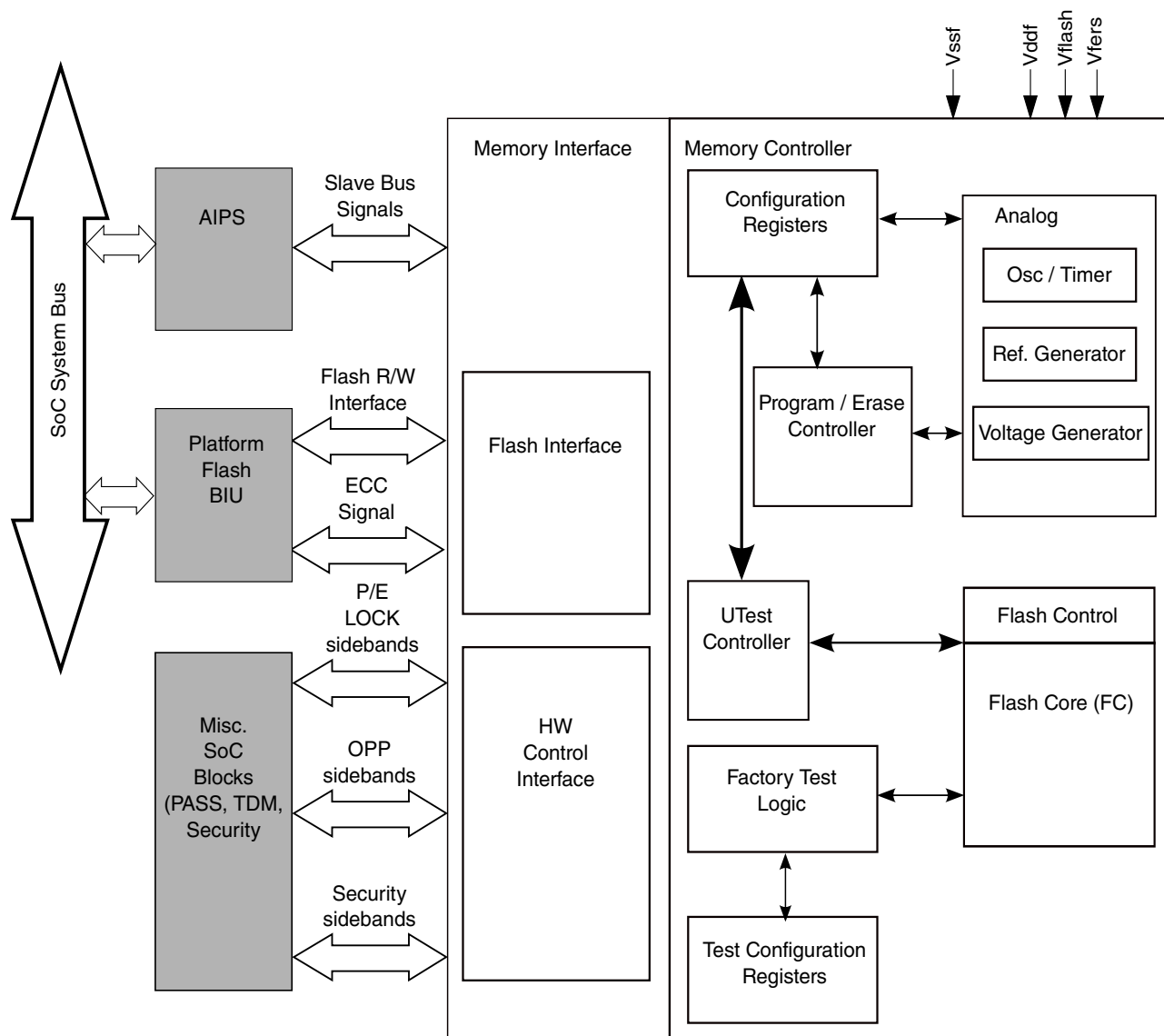


**Figure 26-1. c55fmc block diagram**



## 26.2.1 Overview

The embedded flash memory is addressable by word (32 bits) or double word (64 bits) for program operation, and page (256 bits) for read operation. Multiple word or double-word writes may be done to the flash memory to fill up the program page buffer (256 bits), enabling page programming (256 bits, requiring 4 double-word writes) and quad-page programming (1024 bits, requiring 16 double-word writes). Flash memory reads always return 256 bits, although read page buffering may be performed by the Bus Interface Unit (BIU).

For more details on the embedded flash memory architecture and features, see [Functional Description](#).

## 26.2.2 Features

The embedded flash memory includes these distinct features:

- Test information stored in a dedicated nonvolatile block (referred to as the UTest block)
- OTP space made available in the UTest block
- Read page size of 256 bits (8 words)
- ECC with single-bit correction, double-bit detection (all 1's valid)
- Quad Page programming (64-bit granularity)
- Software programmable block program/erase restriction control
- Erasing of selected block(s)
- Independent programming of the UTest NVM block
- Embedded hardware program and erase algorithms
- Support for reading while writing when the accesses are to different partitions
- Erase suspend, program suspend, and erase-suspended program
- UTest mode (user-accessible test modes) including Array Integrity and Margin Read
- Triple Voted Flops for flash functions requiring high reliability, e.g., internal trimming, redundancy, and mode control

## 26.2.3 Modes of operation

Following is a brief description of the embedded flash memory operating modes.

- *User mode* is the default operating mode of the embedded flash memory. In this mode, it is possible to read and write registers, read and interlock write the memory

array, program the memory array, and erase the memory array. In this mode program and erase operations are initiated by doing array and register writes, and are controlled by an internal state machine.

- *UTest mode* is a tiered test mode strategy in which a portion of the factory test modes are made available. This mode is protected but accessible.

## 26.3 UTest NVM block

The UTest NVM block may be enabled by the BIU. When the UTest NVM space is enabled, all operations are mapped to the UTest NVM block. User-mode programming of the UTest NVM block is enabled only when MCR[PEAS] is high. The UTest NVM block is an OTP block (assuming Test Mode Disable Seal is written) - thus erase is not allowed.

The UTest NVM block supports RWW, and is grouped with the blocks in partition 0.

The UTest NVM block may be locked against program by using the lock registers.

Programming of the UTest NVM space has restrictions that are similar to those of the main space in terms of how ECC is calculated. Only one program operation is allowed per 64-bit ECC segment.

The UTest NVM block contains specified data that is needed for embedded flash memory or SoC features.

## 26.4 Test mode disable seal

Also included in the UTest NVM block is a mechanism to disable factory entry into Test mode. Extreme care must be taken when using this feature, as blocks that are selected to be protected in this method will not be able to have possible failures analyzed by factory failure analysts. Once sealed, the UTest block becomes OTP (Erase Locked, and Over Program Protection enabled). The UTest Block also is not accessible in Test mode.

Protection of this sort prevents all high-voltage operations to the flash executed by the internal state-machine, as well as reads through this state machine, and reads through the Array Integrity state machine when using Test mode interfaces.

UTest operations Margin Read and Array Integrity are also protected by preventing MISR updates on blocks selected for protection, although reads will still be done and single bit corrections and double bit detections will be logged. Protection through other interfaces is protected by normal User mode protection mechanisms, and are device-specific.

The method to disable factory entry into Test mode is to first program the Test Mode Disable Seal location to be 0x2D3C\_4B5A. Once the next reset is asserted, Test mode will be disabled.

It is possible to create a password to enable factory entry into test mode. This can be programmed into the Test Mode Disable Override Passcode. The passcode may not be 0x0000\_0000, 0xFFFF\_FFFF, or 0x5555\_5555. These are all invalid passcodes and will not be accepted to override. If it is desired by the customer that override never be possible, one of the three invalid passcodes should be put into this location. Passcode may be entered to authenticate entry (if enabled) by doing a 32-bit register write to register address 0x90. The UTest NVM block is always protected even if the Test Mode Disable Seal password is written. UTest operations Margin Read and Array Integrity are always protected even if the Test Mode Disable Seal password is written.

Only blocks selected in the Test Mode Disable Block Select field(s) are controlled by the Test Mode Disable feature. Thus it is possible for customers to selectively pick blocks that have this type of protection, and will not be eligible for factory failure analysis. Bits programmed to 0 in the Test Mode Disable Block Select field(s) will designate blocks that are controlled by the Test Mode Disable feature. If either Test Mode Disable Block Select - 0 or Test Mode Disable Block Select - 1 have a 0 programmed, that block will be protected. In order to provide two opportunities to select blocks for the Test Mode Disable, two regions are available, Test Mode Disable Block Select - 0 and Test Mode Disable Block Select - 1. The bit of these two regions are logically ANDed, to define which blocks are effectively selected for the Test Mode Disable feature. The Block Select field is organized as follows, and aligns with how blocks are defined in the LOCK registers.

**Table 26-2. Test mode disable block select**

| Block              | Bits used to select blocks |
|--------------------|----------------------------|
| Low Block Disable  | Data[13:0]                 |
| Unused             | Data[15:14]                |
| Mid Block Disable  | Data[31:16]                |
| High Block Disable | Data[47:32]                |
| 256K Block Disable | Data[95:48]                |

## 26.5 Memory map and register definition

The embedded flash memory map consists of a flash memory array (which includes main array space and UTest NVM space) and a region of registers associated with the programming model that enable flash memory array operation and modification.

The address space consists of 16 KB, 32 KB, 64 KB, and 256 KB blocks.

**C55FMC memory map**

| Address offset (hex) | Register name                                        | Width (in bits) | Access | Reset value                 | Section/ page               |
|----------------------|------------------------------------------------------|-----------------|--------|-----------------------------|-----------------------------|
| 0                    | Module Configuration Register (C55FMC_MCR)           | 32              | R/W    | 0000_0600h                  | <a href="#">26.5.1/882</a>  |
| 8                    | Extended Module Configuration Register (C55FMC_MCRE) | 32              | R      | <a href="#">See section</a> | <a href="#">26.5.2/887</a>  |
| 10                   | Lock 0 register (C55FMC_LOCK0)                       | 32              | R/W    | BFFF_FFFFh                  | <a href="#">26.5.3/890</a>  |
| 14                   | Lock 1 register (C55FMC_LOCK1)                       | 32              | R/W    | 0000_FFFFh                  | <a href="#">26.5.4/892</a>  |
| 18                   | Lock 2 register (C55FMC_LOCK2)                       | 32              | R/W    | FFFF_FFFFh                  | <a href="#">26.5.5/892</a>  |
| 1C                   | Lock 3 register (C55FMC_LOCK3)                       | 32              | R/W    | 0000_FFFFh                  | <a href="#">26.5.6/893</a>  |
| 38                   | Select 0 register (C55FMC_SEL0)                      | 32              | R/W    | 0000_0000h                  | <a href="#">26.5.7/894</a>  |
| 3C                   | Select 1 register (C55FMC_SEL1)                      | 32              | R/W    | 0000_0000h                  | <a href="#">26.5.8/895</a>  |
| 40                   | Select 2 register (C55FMC_SEL2)                      | 32              | R/W    | 0000_0000h                  | <a href="#">26.5.9/896</a>  |
| 44                   | Select 3 register (C55FMC_SEL3)                      | 32              | R/W    | 0000_0000h                  | <a href="#">26.5.10/897</a> |
| 50                   | Address register (C55FMC_ADR)                        | 32              | R      | 0000_0000h                  | <a href="#">26.5.11/898</a> |
| 54                   | UTest 0 register (C55FMC_UT0)                        | 32              | R/W    | 0000_0001h                  | <a href="#">26.5.12/900</a> |
| 58                   | UMISR register (C55FMC_UM0)                          | 32              | R/W    | 0000_0000h                  | <a href="#">26.5.13/903</a> |
| 5C                   | UMISR register (C55FMC_UM1)                          | 32              | R/W    | 0000_0000h                  | <a href="#">26.5.13/903</a> |
| 60                   | UMISR register (C55FMC_UM2)                          | 32              | R/W    | 0000_0000h                  | <a href="#">26.5.13/903</a> |
| 64                   | UMISR register (C55FMC_UM3)                          | 32              | R/W    | 0000_0000h                  | <a href="#">26.5.13/903</a> |
| 68                   | UMISR register (C55FMC_UM4)                          | 32              | R/W    | 0000_0000h                  | <a href="#">26.5.13/903</a> |
| 6C                   | UMISR register (C55FMC_UM5)                          | 32              | R/W    | 0000_0000h                  | <a href="#">26.5.13/903</a> |
| 70                   | UMISR register (C55FMC_UM6)                          | 32              | R/W    | 0000_0000h                  | <a href="#">26.5.13/903</a> |
| 74                   | UMISR register (C55FMC_UM7)                          | 32              | R/W    | 0000_0000h                  | <a href="#">26.5.13/903</a> |

*Table continues on the next page...*

**C55FMC memory map (continued)**

| Address offset (hex) | Register name                                          | Width (in bits) | Access | Reset value                 | Section/ page               |
|----------------------|--------------------------------------------------------|-----------------|--------|-----------------------------|-----------------------------|
| 78                   | UMISR register (C55FMC_UM8)                            | 32              | R/W    | 0000_0000h                  | <a href="#">26.5.13/903</a> |
| 7C                   | UMISR register (C55FMC_UM9)                            | 32              | R/W    | 0000_0000h                  | <a href="#">26.5.14/904</a> |
| 80                   | Over-Program Protection 0 register (C55FMC_OPP0)       | 32              | R      | <a href="#">See section</a> | <a href="#">26.5.15/905</a> |
| 84                   | Over-Program Protection 1 register (C55FMC_OPP1)       | 32              | R      | <a href="#">See section</a> | <a href="#">26.5.16/906</a> |
| 88                   | Over-Program Protection 2 register (C55FMC_OPP2)       | 32              | R      | <a href="#">See section</a> | <a href="#">26.5.17/907</a> |
| 8C                   | Over-Program Protection 3 register (C55FMC_OPP3)       | 32              | R      | <a href="#">See section</a> | <a href="#">26.5.18/907</a> |
| 90                   | Test Mode Disable Password Check register (C55FMC_TMD) | 32              | R/W    | 0000_0000h                  | <a href="#">26.5.19/908</a> |

## 26.5.1 Module Configuration Register (C55FMC\_MCR)

### NOTE

1. A number of Module Configuration Register (MCR) bits are locked against write by other bits. These locks are discussed in relationship to each bit in this section. Simultaneously writing bits which lock each other out is also discussed in [Functional Description](#).
2. See [Functional Description](#) for information about simultaneous MCR writes, and priority levels.

Address: 0h base + 0h offset = 0h

|       |     |     |     |     |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|-----|-----|-----|-----|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0   | 1   | 2   | 3   | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | RVE | RRE | AEE | EEE | 0 |   |   |   |   |   |    |    |    |    |    |    |
| W     | w1c | w1c | w1c | w1c |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0   | 0   | 0   | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |     |     |     |    |      |      |     |       |    |    |    |    |     |      |     |      |     |
|-------|-----|-----|-----|----|------|------|-----|-------|----|----|----|----|-----|------|-----|------|-----|
| Bit   | 16  | 17  | 18  | 19 | 20   | 21   | 22  | 23    | 24 | 25 | 26 | 27 | 28  | 29   | 30  | 31   |     |
| R     | EER | RWE | SBC | 0  | PEAS | DONE | PEG | PECIE | 0  |    |    |    | PGM | PSUS | ERS | ESUS | EHV |
| W     | w1c | w1c | w1c |    |      |      |     |       |    |    |    |    |     |      |     |      |     |
| Reset | 0   | 0   | 0   | 0  | 0    | 1    | 1   | 0     | 0  | 0  | 0  | 0  | 0   | 0    | 0   | 0    |     |

### C55FMC\_MCR field descriptions

| Field    | Description                                                                                                                                                                                                                        |
|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>RVE | Read Voltage Error<br><br>RVE provides information on previous reads monitoring the read pump voltage. If the read voltage is detected to be out of range, this bit is set to indicate that previous reads requested may have been |

Table continues on the next page...

**C55FMC\_MCR field descriptions (continued)**

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                  | <p>corrupted. Since this bit is a status flag, it must be cleared to a 0 by writing a 1 to the register location. A write of 0 has no effect.</p> <p>0 Reads are occurring without voltage issues</p> <p>1 A previous read may have been corrupted due to read voltage being out of range</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 1<br>RRE         | <p>Read Reference Error</p> <p>RRE provides information on previous reads monitoring the read reference. If the read reference is detected to be out of range, this bit is set to indicate that previous reads requested may have been corrupted. Since this bit is a status flag, it must be cleared to a 0 by writing a 1 to the register location. A write of 0 has no effect.</p> <p>0 Reads are occurring without reference issues</p> <p>1 A previous read may have been corrupted due to read reference being out of range</p>                                                                                                                                                                                                                                                                                                                                                     |
| 2<br>AEE         | <p>Address Encode Error</p> <p>AEE provides information on previous reads monitoring the address encode feature. On every read request to the flash, the incoming address is compared to an encoded address (row, column, and block) coming back from the memory array using the read data sense amplifier timing. If these two values do not match (zero selected, multiple selected, wrong selected), or the timing is incorrect, an address encode error will be recorded. If an address encode mismatch is detected, this bit is set to indicate that previous reads requested may have been corrupted. Since this bit is a status flag, it must be cleared to a 0 by writing a 1 to the register location. A write of 0 has no effect.</p> <p>0 Reads are occurring without address encode mismatches</p> <p>1 A previous read may be corrupted based on address encode mismatch</p> |
| 3<br>EEE         | <p>ECC after ECC Error</p> <p>EEE provides information on previous reads monitoring the ECC after ECC feature. On every read request to the flash, ECC is recalculated serially, and if there is a mismatch between the ECC calculations (taking into account corrections or detections) a late error will be reported. If an ECC after ECC mismatch is detected, this bit is set to indicate that previous reads requested may have been corrupted. Since this bit is a status flag, it must be cleared to a 0 by writing a 1 to the register location. A write of 0 has no effect.</p> <p>0 Reads are occurring without ECC after ECC mismatches</p> <p>1 A previous read may be corrupted based on ECC calculation errors</p>                                                                                                                                                          |
| 4–15<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 16<br>EER        | <p>ECC Event Error</p> <p>This bit provides information on previous reads. If a double bit detection occurred, the EER bit is set to a '1'. This bit must then be cleared, or a reset must occur before it returns to a 0 state. This bit may not be set by software. In the event of a single bit detection and correction, this bit is not set. If EER is not set, or remains 0, this indicates that all previous reads (from the last reset, or clearing of EER) are correct. Since this bit is an error flag, it must be cleared to a 0 by writing a 1 to the register location. A write of 0 has no effect.</p> <p>0 Reads are occurring normally.</p> <p>1 An ECC Error occurred during a previous read.</p>                                                                                                                                                                        |
| 17<br>RWE        | <p>Read-While-Write Event Error</p> <p>This bit provides information on previous read-while-write (RWW) reads. If an RWW error occurs, this bit is set to 1. The bit must then be cleared, or a reset must occur before it returns to a 0 state. This bit may not be written to a 1. If RWE is not set, or remains 0, this indicates that all previous RWW reads (from the last</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |

*Table continues on the next page...*

**C55FMC\_MCR field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | <p>reset, or clearing of RWE) are correct. Since this bit is an error flag, it must be cleared to a 0 by writing a 1 to the register location. A write of 0 has no effect.</p> <p>0 Reads are occurring normally.<br/>1 A read-while-write error occurred during a previous read.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 18<br>SBC      | <p>Single Bit Correction</p> <p>SBC provides information on previous reads, if the SBCE is set. If a single bit correction occurred, the SBC bit is set to a 1. This bit must then be cleared, or a reset must occur before this bit returns to a 0 state. If SBC is not set, or remains 0, this indicates that all previous reads (from the last reset, or clearing of SBC) did not require a correction. Since this bit is a status flag, it must be cleared to a 0 by writing a 1 to the register location. A write of 0 has no effect.</p> <p>0 Reads are occurring without corrections.<br/>1 A Single Bit Correction occurred during a previous read.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 19<br>Reserved | <p>This field is reserved.<br/>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 20<br>PEAS     | <p>Program Access Space</p> <p>PEAS indicates which space is valid for program or erase operations — either main array space or UTest NVM space. It should be noted that if the flash is sealed, and the Test Mode Disable Seal is programmed, erase is not allowed on the UTest NVM space. PEAS = 0 indicates the main address space is active for all program or erase operations. PEAS = 1 indicates the UTest NVM address space is active for program or erase (if applicable) operations. The value in PEAS is captured and held when the UTest NVM block is enabled with the first interlock write done for program operations (and erase operations if the flash is unsealed). The value of PEAS is retained between sampling events (in other words, subsequent first interlock writes). The value in PEAS may be changed during erase-suspended program, and reverts back to its original state once the erase-suspended program is completed. PEAS is read only.</p> <p>0 UTest NVM address space is disabled for program/erase and main address space enabled.<br/>1 UTest NVM address space is enabled for program/erase and main address space disabled.</p> |
| 21<br>DONE     | <p>State Machine Status</p> <p>DONE indicates whether the embedded flash memory is performing a high voltage operation. DONE is set to a 1 on termination of the embedded flash memory reset. DONE is read only. DONE is set to a 1 at the end of program and erase high voltage sequences.</p> <p><b>NOTE:</b> This bit transitions from a 0 to 1 during reset and remains at 1 after reset.</p> <p>0 Flash memory is executing a high voltage operation.<br/>1 Flash memory is not executing a high voltage operation.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 22<br>PEG      | <p>Program/Erase Good</p> <p>The PEG bit indicates the completion status of the last flash memory program or erase sequence for which high voltage operations were initiated. The value of PEG is updated automatically during the program and erase high voltage operations. Aborting a program/erase high voltage operation causes PEG to be cleared, indicating the sequence failed. PEG is set to a 1 when the embedded flash memory is reset. PEG is read only.</p> <p>The value of PEG is valid only when PGM = 1 or ERS = 1 and after DONE transitions from 0 to 1 due to an abort or the completion of a program/erase operation. PEG is valid until PGM/ERS makes a 1 to 0 transition or EHV makes a 0 to 1 transition. The value in PEG is not valid after a 0 to 1 transition of DONE caused by PSUS or ESUS being set to logic 1. If PGM and ERS are both 1 when DONE makes a qualifying 0 to 1 transition the value of PEG indicates the completion status of the PGM sequence. This happens in an erase-suspended program operation.</p>                                                                                                                    |

*Table continues on the next page...*



**C55FMC\_MCR field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | <p>PEG will be 0 if an Erase-Suspended Program is attempted to a block that has been selected for Erase.</p> <p>PEG will also be 0 in the event of an error in the interlock write for program only. An error in the interlock write occurs when the ECC value provided to the flash memory does not match the ECC calculated within the flash memory. In the event of this, high voltage will not be applied to the array, and PEG will indicate a failed program.</p> <p><b>NOTE:</b></p> <ol style="list-style-type: none"> <li>1. If program or erase operations are attempted on blocks that are locked, the response from embedded flash memory is PEG = 1, indicating that the operation was successful, and the contents of the block are properly protected from the program or erase operation. PEG = 1 is also true if an abort occurs during an HV request to a locked block.</li> <li>2. If a program operation is attempted to a location marked as OTP, the response from the embedded flash memory is PEG = 0, indicating that the operation was not allowed, and the value interlocked was not programmed, since the desired doubleword was already programmed with a previous program operation.</li> </ol> <p>0 Program or erase operation failed.<br/>1 Program or erase operation successful.</p> |
| 23<br>PECIE       | <p>Program/Erase Complete Interrupt Enable</p> <p>PECIE provides a mechanism to trigger an interrupt request upon the assertion of the DONE flag due to a high voltage event (program or erase) finishing (normal, abort, suspend). If PECIE is written while not in a high voltage event, the interrupt will not immediately trigger, but will trigger after the next high voltage event is completed.</p> <p>0 An interrupt request will not be generated when the DONE flag is set.<br/>1 An interrupt request will be generated when the DONE flag is set.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 24–26<br>Reserved | <p>This field is reserved.<br/>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 27<br>PGM         | <p>Program</p> <p>PGM is used to set up embedded flash memory for a program operation. A 0 to 1 transition of PGM initiates a program sequence. A 1 to 0 transition of PGM ends the program sequence. PGM can be set only under one of the following conditions:</p> <ul style="list-style-type: none"> <li>• User mode read (ERS is low and UTE is low)</li> <li>• Erase suspend (ERS and ESUS are 1) with EHV low</li> </ul> <p>PGM can be cleared only when PSUS and EHV are low and DONE is high. PGM is cleared on reset.</p> <p>0 Flash memory is not executing a program sequence.<br/>1 Flash memory is executing a program sequence.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 28<br>PSUS        | <p>Program Suspend</p> <p>PSUS is used to indicate the embedded flash memory is in program suspend or in the process of entering a suspend state. The embedded flash memory is in program suspend when PSUS = 1 and DONE = 1. PSUS can be set high only when PGM and EHV are high. A 0 to 1 transition of PSUS starts the sequence which sets DONE and places the embedded flash memory in program suspend. The embedded flash memory enters suspend within Tpsus of this transition.</p> <p>PSUS can be cleared only when DONE and EHV are high. A 1 to 0 transition of PSUS with EHV = 1 starts the sequence which clears DONE and returns the embedded flash memory to program. The embedded flash memory cannot exit program suspend and clear DONE while EHV is low. PSUS is cleared on reset.</p> <p>0 Program sequence is not suspended.<br/>1 Program sequence is suspended.</p>                                                                                                                                                                                                                                                                                                                                                                                                                               |

*Table continues on the next page...*

**C55FMC\_MCR field descriptions (continued)**

| Field      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 29<br>ERS  | <p>Erase</p> <p>ERS is used to set up embedded flash memory for an erase operation. A 0 to 1 transition of ERS initiates an erase sequence. A 1 to 0 transition of ERS ends the erase sequence. ERS can only be set in user mode read (PGM is low and UTE is low). ERS can be cleared only when ESUS and EHV are low and DONE is high. ERS is cleared on reset.</p> <p>0 Flash memory is not executing an erase sequence.<br/>1 Flash memory is executing an erase sequence.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 30<br>ESUS | <p>Erase Suspend</p> <p>ESUS is used to indicate that the embedded flash memory is in erase suspend or in the process of entering a suspend state. The embedded flash memory is in erase suspend when ESUS = 1 and DONE = 1. ESUS can be set high only when ERS and EHV are high and PGM is low. A 0 to 1 transition of ESUS starts the sequence which sets DONE and places the flash memory in erase suspend. The embedded flash memory enters suspend within T<sub>es</sub> of this transition.</p> <p>ESUS can be cleared only when DONE and EHV are high and PGM is low. A 1 to 0 transition of ESUS with EHV = 1 starts the sequence which clears DONE and returns the embedded flash memory to erase. The embedded flash memory cannot exit erase suspend and clear DONE while EHV is low. ESUS is cleared on reset.</p> <p>0 Erase sequence is not suspended.<br/>1 Erase sequence is suspended.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 31<br>EHV  | <p>Enable High Voltage</p> <p>The EHV bit enables the embedded flash memory for a high voltage program/erase operation. EHV is cleared on reset. EHV must be set after an interlock write to start a program/erase sequence. EHV may be set, initiating a program/erase, after an interlock under one of the following conditions:</p> <ul style="list-style-type: none"> <li>• Erase (ERS = 1, ESUS = 0)</li> <li>• Program (ERS = 0, ESUS = 0, PGM = 1, PSUS = 0)</li> <li>• Erase-suspended program (ERS = 1, ESUS = 1, PGM = 1, PSUS = 0)</li> </ul> <p>If a program operation is to be initiated while an erase is suspended, EHV must be cleared while in erase suspend before setting PGM.</p> <p>In normal operation, a 1 to 0 transition of EHV with DONE high, PSUS low, and ESUS low terminates the current program/erase high-voltage operation. In an erase-suspended program operation, a 1 to 0 transition of EHV with DONE high, ESUS high, PGM high, ERS high, and PSUS low terminates the program that is the current high voltage operation.</p> <p>When an operation is aborted, there is a 1 to 0 transition of EHV with DONE low and the suspend bit for the current program/erase sequence low. An abort causes the value of PEG to be cleared, indicating a failed program/erase; address locations being operated on by the aborted operation contain indeterminate data after an abort. EHV may not be written to a 1 after an abort is requested (EHV being cleared) and before DONE transitions high.</p> <p>A suspended operation cannot be aborted. EHV may be written during suspend. EHV must be high for the embedded flash memory to exit suspend. EHV may not be written after a suspend bit is set high and before DONE transitions high. EHV may not be set low after the current suspend bit is set low and before DONE transitions low.</p> <p><b>NOTE:</b> Aborting a high voltage operation leaves FC addresses in an indeterminate data state. This may be recovered by executing an erase on the affected blocks.</p> <p>0 Flash memory is not enabled to perform a high voltage operation.<br/>1 Flash memory is enabled to perform a high voltage operation.</p> |

## 26.5.2 Extended Module Configuration Register (C55FMC\_MCRE)

### NOTE

The reset value of this register is set at the factory and varies among devices. See the chip-specific C55FMC information for details.

Address: 0h base + 8h offset = 8h

| Bit   | 0  | 1  | 2     | 3  | 4  | 5  | 6  | 7  | 8     | 9  | 10 | 11    | 12 | 13    | 14 | 15 |
|-------|----|----|-------|----|----|----|----|----|-------|----|----|-------|----|-------|----|----|
| R     | HT | 0  | n256K |    |    |    |    |    | n64Kh |    |    | n32Kh |    | n16Kh |    |    |
| W     |    |    |       |    |    |    |    |    |       |    |    |       |    |       |    |    |
| Reset | 0* | 0* | 0*    | 0* | 0* | 0* | 0* | 0* | 0*    | 0* | 0* | 0*    | 0* | 0*    | 0* | 0* |

| Bit   | 16    | 17 | 18 | 19    | 20 | 21    | 22 | 23 | 24    | 25 | 26 | 27    | 28 | 29    | 30 | 31 |
|-------|-------|----|----|-------|----|-------|----|----|-------|----|----|-------|----|-------|----|----|
| R     | n64Km |    |    | n32Km |    | n16Km |    |    | n64Kl |    |    | n32Kl |    | n16Kl |    |    |
| W     |       |    |    |       |    |       |    |    |       |    |    |       |    |       |    |    |
| Reset | 0*    | 0* | 0* | 0*    | 0* | 0*    | 0* | 0* | 0*    | 0* | 0* | 0*    | 0* | 0*    | 0* | 0* |

\* Notes:

- 

### C55FMC\_MCRE field descriptions

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>HT         | <p>High Temperature Enabled.</p> <p>HT provides configuration information of the module. If the flash is tuned for high temperature operation (165 °C) at reduced performance (slower read at 165 °C and less and slower program and erase at 150 °C or less), and limited features at this temperature (no margin reads), the HT status bit will be a 1. Default is 150 °C as maximum operating temperature, and the HT status bit will be a 0. HT is read only.</p> <p>0 150 °C maximum operating temperature.</p> <p><b>NOTE:</b> 165 °C operating temperature maximum is not supported on this device.</p> <p>1 Reserved</p> |
| 1–2<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 3–7<br>n256K    | <p>Number of 256 KB blocks. The value of this field is dependent upon the size of the embedded flash memory. This field is read only.</p> <p>00000 Zero 256 KB blocks.</p> <p>00001 Two 256 KB blocks (512 KB total).</p> <p>00010 Four 256 KB blocks (1 MB total).</p> <p>00011 Six 256 KB blocks (1.5 MB total).</p> <p>00100 Eight 256 KB blocks (2 MB total).</p> <p>00101 Ten 256 KB blocks (2.5 MB total).</p>                                                                                                                                                                                                             |

Table continues on the next page...

**C55FMC\_MCRE field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | 00110 Twelve 256 KB blocks (3 MB total).<br>00111 Fourteen 256 KB blocks (3.5 MB total).<br>01000 Sixteen 256 KB blocks (4 MB total).<br>01001 Eighteen 256 KB blocks (4.5 MB total).<br>01010 Twenty 256 KB blocks (5 MB total).<br>01011 Twenty-two 256 KB blocks (5.5 MB total).<br>01100 Twenty-four 256 KB blocks (6 MB total).<br>01101 Twenty-six 256 KB blocks (6.5 MB total).<br>01110 Twenty-eight 256 KB blocks (7 MB total).<br>01111 Thirty 256 KB blocks (7.5 MB total).<br>10000 Thirty-two 256 KB blocks (8 MB total).<br>10001 Reserved<br>10010 Reserved<br>10011 Reserved<br>10100 Reserved<br>10101 Reserved<br>10110 Reserved<br>10111 Reserved<br>11000 Forty-eight 256 KB blocks (12 MB total).<br>11001 Reserved.<br>11010 Reserved.<br>11011 Reserved.<br>11100 Reserved.<br>11101 Reserved.<br>11110 Reserved.<br>11111 Reserved. |
| 8–10<br>n64Kh  | Number of 64 KB blocks in partitions 4 and 5. This field is read only.<br>000 Zero 64 KB blocks.<br>001 Two 64 KB blocks.<br>010 Four 64 KB blocks.<br>011 Six 64 KB blocks.<br>100 Eight 64 KB blocks.<br>101 Twelve 64 KB blocks..<br>110 Sixteen 64 KB blocks.<br>111 Reserved.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 11–12<br>n32Kh | Number of 32 KB blocks in partitions 4 and 5. This field is read only.<br>00 Zero 32 KB blocks.<br>01 Two 32 KB blocks.<br>10 Four 32 KB blocks.<br>11 Reserved.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 13–15<br>n16Kh | Number of 16 KB blocks in partitions 4 and 5. This field is read only.<br>000 Zero 16 KB blocks.<br>001 Two 16 KB blocks.<br>010 Four 16 KB blocks.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

*Table continues on the next page...*

**C55FMC\_MCRE field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                    |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | 011 Six 16 KB blocks.<br>100 Eight 16 KB blocks.<br>101 Reserved.<br>110 Reserved.<br>111 Reserved.                                                                                                                                                            |
| 16–18<br>n64Km | Number of 64 KB blocks in partitions 2 and 3. This field is read only.<br><br>000 Zero 64 KB blocks.<br>001 Two 64 KB blocks.<br>010 Four 64 KB blocks.<br>011 Six 64 KB blocks.<br>100 Eight 64 KB blocks.<br>101 Reserved.<br>110 Reserved.<br>111 Reserved. |
| 19–20<br>n32Km | Number of 32 KB blocks in partitions 2 and 3. This field is read only.<br><br>00 Zero 32 KB blocks.<br>01 Two 32 KB blocks.<br>10 Four 32 KB blocks.<br>11 Reserved.                                                                                           |
| 21–23<br>n16Km | Number of 16 KB blocks in partitions 2 and 3. This field is read only.<br><br>000 Zero 16 KB blocks.<br>001 Two 16 KB blocks.<br>010 Four 16 KB blocks.<br>011 Six 16 KB blocks.<br>100 Eight 16 KB blocks.<br>101 Reserved.<br>110 Reserved.<br>111 Reserved. |
| 24–26<br>n64KI | Number of 64 KB blocks in partitions 0 and 1. This field is read only.<br><br>000 Zero 64 KB blocks.<br>001 Two 64 KB blocks.<br>010 Four 64 KB blocks.<br>011 Six 64 KB blocks.<br>100 Eight 64 KB blocks.<br>101 Reserved.<br>110 Reserved.<br>111 Reserved. |
| 27–28<br>n32KI | Number of 32 KB blocks in partitions 0 and 1. This field is read only.<br><br>00 Zero 32 KB blocks.<br>01 Two 32 KB blocks.<br>10 Four 32 KB blocks.<br>11 Reserved.                                                                                           |

*Table continues on the next page...*

**C55FMC\_MCRE field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                          |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 29–31<br>n16KI | Number of 16 KB blocks in partitions 0 and 1. This field is read only.<br><br>000 Zero 16 KB blocks.<br>001 Two 16 KB blocks.<br>010 Four 16 KB blocks.<br>011 Six 16 KB blocks.<br>100 Reserved.<br>101 Reserved.<br>110 Reserved.<br>111 Reserved. |

**26.5.3 Lock 0 register (C55FMC\_LOCK0)**

The Lock 0 (LOCK0) register provides a means to protect blocks from being modified.

Address: 0h base + 10h offset = 10h

|       |         |    |         |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---------|----|---------|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0       | 1  | 2       | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | TSLOCK  | 0  | LOWLOCK |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |         |    |         |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 1       | 0  | 1       | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  |
| Bit   | 16      | 17 | 18      | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | MIDLOCK |    |         |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |         |    |         |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 1       | 1  | 1       | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  |

**C55FMC\_LOCK0 field descriptions**

| Field       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>TSLOCK | <p>UTest NVM Lock.</p> <p>This bit is used to lock the UTest NVM block from programs (erase protection not needed since UTest NVM is OTP and not erasable). A value of 1 in the TSLOCK register signifies that the UTest NVM block is locked for program. A value of 0 in the TSLOCK register signifies that the UTest NVM block is available to receive program pulses.</p> <p>The TSLOCK register is not writable once an interlock write is completed until DONE is set at the completion of the requested operation. Likewise, TSLOCK register is not writable if a high voltage operation is suspended.</p> |

*Table continues on the next page...*

**C55FMC\_LOCK0 field descriptions (continued)**

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                  | 0 UTest NVM block is available to be programmed.<br>1 UTest NVM block is locked and cannot be programmed.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 1<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 2–15<br>LOWLOCK  | <b>Low Block Lock</b><br>A value of 1 in a bit of the lock register signifies that the corresponding block is locked for program and erase. A value of 0 in the lock register signifies that the corresponding block is available to receive program and erase pulses. The block numbering for the Low Blocks starts at LOWLOCK[0] with 16 KB block region (if available), then the 32 KB block region (if available), and lastly the 64 KB block region (if available). This continues until all blocks are accounted for.<br>Low Blocks exist in partitions 0 and 1.<br>The lock register is not writable once an interlock write is completed until DONE is set at the completion of the requested operation. Likewise, the lock register is not writable if a high voltage operation is suspended.<br>The default value of the LOCK bits is locked.<br>In the event that blocks are not present (due to configuration or total memory size), the LOCK bits default to be locked, and are not writable. The reset value is always 1, and register writes have no effect.<br>0 Block is available for program and erase operations.<br>1 Block is locked and unavailable for program and erase operations |
| 16–31<br>MIDLOCK | <b>Mid Block Lock</b><br>A value of 1 in a bit of the lock register signifies that the corresponding block is locked for program and erase. A value of 0 in the lock register signifies that the corresponding block is available to receive program and erase pulses. The block numbering for the Mid Blocks starts at MIDLOCK[0] with 16 KB block region (if available), then the 32 KB block region (if available), and lastly the 64 KB block region (if available). This continues until all blocks are accounted for.<br>Mid Blocks exist in partitions 2 and 3.<br>The lock register is not writable once an interlock write is completed until DONE is set at the completion of the requested operation. Likewise, the lock register is not writable if a high-voltage operation is suspended.<br>The default value of the LOCK bits is locked.<br>In the event that blocks are not present (due to configuration or total memory size), the LOCK bits default to be locked, and are not writable. The reset value is always 1, and register writes have no effect.<br>0 Block is available for program and erase operations.<br>1 Block is locked and unavailable for program and erase operations |

## 26.5.4 Lock 1 register (C55FMC\_LOCK1)

The Lock 1 (LOCK1) register provides a means to protect blocks from being modified.

Address: 0h base + 14h offset = 14h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16       | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | HIGHLOCK |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 1        | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  |    |

### C55FMC\_LOCK1 field descriptions

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 16–31<br>HIGHLOCK | <p>High Block Lock</p> <p>A value of 1 in a bit of the lock register signifies that the corresponding block is locked for program and erase. A value of 0 in the lock register signifies that the corresponding block is available to receive program and erase pulses. The block numbering for the High Blocks starts at HIGHLOCK[0] with 16 KB block region (if available), then the 32 KB block region (if available), and lastly the 64 KB block region (if available). This continues until all blocks are accounted.</p> <p>HIGH Blocks exist in partitions 4 and 5.</p> <p>The lock register is not writable once an interlock write is completed until DONE is set at the completion of the requested operation. Likewise, the lock register is not writable if a high-voltage operation is suspended.</p> <p>The default value of the LOCK bits is locked.</p> <p>In the event that blocks are not present (due to configuration or total memory size), the LOCK bits default to be locked, and are not writable. The reset value is always 1, and register writes have no effect.</p> <p>0 Block is available for program and erase operations.<br/>1 Block is locked and unavailable for program and erase operations</p> |

## 26.5.5 Lock 2 register (C55FMC\_LOCK2)

The Lock 2 (LOCK2) register provides a means to protect blocks from being modified.

Address: 0h base + 18h offset = 18h

|       |           |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-----------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0         | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |           |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |           |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|       | A256KLOCK |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 1         | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  |    |



**C55FMC\_LOCK2 field descriptions**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>A256KLOCK | <p>256 KB Block Lock</p> <p>256KLOCK has the same characteristics as LOWLOCK. Please see that description for more information. The block numbering for the 256 KB blocks starts with 256KLOCK[0] and continues until all blocks are accounted.</p> <p>A value of 1 in a bit of the lock register signifies that the corresponding block is locked for program and erase. A value of 0 in the lock register signifies that the corresponding block is available to receive program and erase pulses. The block numbering for the 256 KB blocks starts with 256KLOCK[0] and continues until all blocks are accounted.</p> <p>The lock register is not writable once an interlock write is completed until DONE is set at the completion of the requested operation. Likewise, the lock register is not writable if a high-voltage operation is suspended.</p> <p>The default value of the LOCK bits is locked.</p> <p>In the event that blocks are not present (due to configuration or total memory size), the LOCK bits default to be locked, and are not writable. The reset value is always 1, and register writes have no effect.</p> <p>0 Block is available for program and erase operations.<br/>1 Block is locked and unavailable for program and erase operations</p> |

**26.5.6 Lock 3 register (C55FMC\_LOCK3)**

The Lock 3 (LOCK3) register provides a means to protect blocks from being modified.

Address: 0h base + 1Ch offset = 1Ch

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|-----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16        | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | A256KLOCK |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 1         | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  |    |

**C55FMC\_LOCK3 field descriptions**

| Field              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>Reserved   | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 16–31<br>A256KLOCK | <p>256 KB Block Lock</p> <p>256KLOCK has the same characteristics as LOWLOCK. Please see that description for more information. The block numbering for the 256 KB blocks starts with 256KLOCK[0] and continues until all blocks are accounted.</p> <p>A value of 1 in a bit of the lock register signifies that the corresponding block is locked for program and erase. A value of 0 in the lock register signifies that the corresponding block is available to receive program and erase pulses. The block numbering for the 256 KB blocks starts with 256KLOCK[0] and continues until all blocks are accounted.</p> |

*Table continues on the next page...*

**C55FMC\_LOCK3 field descriptions (continued)**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | <p>The lock register is not writable once an interlock write is completed until DONE is set at the completion of the requested operation. Likewise, the lock register is not writable if a high-voltage operation is suspended.</p> <p>The default value of the LOCK bits is locked.</p> <p>In the event that blocks are not present (due to configuration or total memory size), the LOCK bits default to be locked, and are not writable. The reset value is always 1, and register writes have no effect.</p> <p>0 Block is available for program and erase operations.<br/> 1 Block is locked and unavailable for program and erase operations</p> |

**26.5.7 Select 0 register (C55FMC\_SEL0)**

The Select 0 (SEL0) register provides a means to select blocks to be operated on during erase.

Address: 0h base + 38h offset = 38h

|       |   |   |        |   |   |   |   |   |   |   |    |    |    |    |    |    |        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|--------|---|---|---|---|---|---|---|----|----|----|----|----|----|--------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2      | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16     | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   | LOWSEL |   |   |   |   |   |   |   |    |    |    |    |    |    | MIDSEL |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |        |   |   |   |   |   |   |   |    |    |    |    |    |    |        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0      | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0      | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**C55FMC\_SEL0 field descriptions**

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–1<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 2–15<br>LOWSEL  | <p>LOW Block Select.</p> <p>A value of 1 in the select register signifies that the block is selected for erase. A value of 0 in the select register signifies that the block is not selected. The reset value for the select registers is 0, or unselected. The block numbering for the Low Blocks starts at LOWSEL[0] with 16 KB block region (if available), then the 32 KB block region (if available), and lastly the 64 KB block region (if available). This continues until all blocks are accounted.</p> <p>Low Blocks exist in partitions 0 and 1.</p> <p>The blocks must be selected (or unselected) before doing an erase interlock write as part of the erase sequence. The select register is not writable once an interlock write is completed until DONE is set at the completion of the requested operation, or if a high-voltage operation is suspended. LOWSEL is also not writeable during UTest operations, when AIE is high.</p> <p>In the event that blocks are not present (due to configuration or total memory size), the corresponding select bits default to unselected, and are not writable. The reset value is always 0, and register writes have no effect.</p> <p>0 The block is not selected<br/> 1 The block is selected for erase</p> |

*Table continues on the next page...*

**C55FMC\_SEL0 field descriptions (continued)**

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 16–31<br>MIDSEL | <p>Mid Block Select.</p> <p>A value of 1 in the select register signifies that the block is selected for erase. A value of 0 in the select register signifies that the block is not selected. The reset value for the select registers is 0, or unselected. The block numbering for the Mid Blocks starts at MIDLSEL[0] with 16 KB block region (if available), then the 32 KB block region (if available), and lastly the 64 KB block region (if available). This continues until all blocks are accounted.</p> <p>Mid Blocks exist in partitions 2 and 3.</p> <p>The blocks must be selected (or unselected) before doing an erase interlock write as part of the erase sequence. The select register is not writable once an interlock write is completed until DONE is set at the completion of the requested operation, or if a high-voltage operation is suspended. MIDSEL is also not writable during UTest operations, when AIE is high.</p> <p>In the event that blocks are not present (due to configuration or total memory size), the corresponding select bits default to unselected, and are not writable. The reset value is always 0, and register writes have no effect.</p> <p>0 The block is not selected<br/>1 The block is selected for erase</p> |

**26.5.8 Select 1 register (C55FMC\_SEL1)**

The Select 1 (SEL1) register provides a means to select blocks to be operated on during erase.

Address: 0h base + 3Ch offset = 3Ch

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|---------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16      | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | HIGHSEL |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0       | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**C55FMC\_SEL1 field descriptions**

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 16–31<br>HIGHSEL | <p>High Block Select.</p> <p>A value of 1 in the select register signifies that the block is selected for erase. A value of 0 in the select register signifies that the block is not selected. The reset value for the select registers is 0, or unselected. The block numbering for the High Blocks starts at HIGHSEL[0] with 16 KB block region (if available), then the 32 KB block region (if available), and lastly the 64 KB block region (if available). This continues until all blocks are accounted.</p> <p>HIGH Blocks exist in partitions 4 and 5.</p> |

*Table continues on the next page...*

**C55FMC\_SEL1 field descriptions (continued)**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | <p>The blocks must be selected (or unselected) before doing an erase interlock write as part of the erase sequence. The select register is not writable once an interlock write is completed until DONE is set at the completion of the requested operation, or if a high-voltage operation is suspended. HIGHSEL is also not writeable during UTest operations, when AIE is high.</p> <p>In the event that blocks are not present (due to configuration or total memory size), the corresponding select bits default to unselected, and are not writable. The reset value is always 0, and register writes have no effect.</p> <p>0 The block is not selected for erase<br/>1 The block is selected for erase</p> |

**26.5.9 Select 2 register (C55FMC\_SEL2)**

The Select 2 (SEL2) register provides a means to select blocks to be operated on during erase.

Address: 0h base + 40h offset = 40h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**C55FMC\_SEL2 field descriptions**

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>A256KSEL | <p>256 KB Block Select.</p> <p>A value of 1 in the select register signifies that the block is selected for erase. A value of 0 in the select register signifies that the block is not selected. The reset value for the select registers is 0, or unselected. The block numbering for the 256 KB blocks starts with 256KSEL[0] and continues until all blocks are accounted.</p> <p>The blocks must be selected (or unselected) before doing an erase interlock write as part of the erase sequence. The select register is not writable once an interlock write is completed until DONE is set at the completion of the requested operation, or if a high-voltage operation is suspended.</p> <p>256KSEL is also not writeable during UTest operations, when AIE is high.</p> <p>In the event that blocks are not present (due to configuration or total memory size), the corresponding select bits default to unselected, and are not writable. The reset value is always 0, and register writes have no effect.</p> <p>0 The block is not selected<br/>1 The block is selected for erase</p> |

## 26.5.10 Select 3 register (C55FMC\_SEL3)

The Select 3 (SEL3) register provides a means to select blocks to be operated on during erase.

Address: 0h base + 44h offset = 44h

|       |          |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0        | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16       | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | Reserved |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | A256KSEL |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |          |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**C55FMC\_SEL3 field descriptions**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>Reserved  | This field is reserved.<br>Reserved, reset to 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 16–31<br>A256KSEL | <p>256 KB Block Select.</p> <p>A value of 1 in the select register signifies that the block is selected for erase. A value of 0 in the select register signifies that the block is not selected. The reset value for the select registers is 0, or unselected. The block numbering for the 256 KB blocks starts with 256KSEL[0] and continues until all blocks are accounted.</p> <p>The blocks must be selected (or unselected) before doing an erase interlock write as part of the erase sequence. The select register is not writable once an interlock write is completed until DONE is set at the completion of the requested operation, or if a high-voltage operation is suspended. 256KSEL is also not writeable during UTest operations, when AIE is high.</p> <p>In the event that blocks are not present (due to configuration or total memory size), the corresponding select bits default to unselected, and are not writable. The reset value is always 0, and register writes have no effect.</p> <p>1 The block is selected for erase.<br/>0 The block is not selected.</p> |

## 26.5.11 Address register (C55FMC\_ADR)

The Address register (ADR) provides the first failing address in the event of a failure (ECC or PGM/Erase state machine), as well as single bit correction information.

Address: 0h base + 50h offset = 50h

|       |     |    |    |    |       |      |      |      |      |   |    |    |    |    |    |    |
|-------|-----|----|----|----|-------|------|------|------|------|---|----|----|----|----|----|----|
| Bit   | 0   | 1  | 2  | 3  | 4     | 5    | 6    | 7    | 8    | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | SAD | aH | aM | aL | a256k | a64k | a32k | a16k | ADDR |   |    |    |    |    |    |    |
| W     |     |    |    |    |       |      |      |      |      |   |    |    |    |    |    |    |
| Reset | 0   | 0  | 0  | 0  | 0     | 0    | 0    | 0    | 0    | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 16   | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | ADDR |    |    |    |    |    |    |    |    |    |    |    |    | 0  |    |    |
| W     |      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

### C55FMC\_ADR field descriptions

| Field    | Description                                                                                                                                                                                                                                                                                                                                                                    |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>SAD | <p>UTest NVM Address. The SAD bit qualifies the address captured during an ECC Event Error, Single Bit Correction, or State Machine operation. SAD has the same characteristics as ADDR related to capturing of addresses.</p> <p>The SAD field is not writable.</p> <p>0 Address captured is from main array Space.<br/>1 Address captured is from UTest NVM array space.</p> |
| 1<br>aH  | <p>Address High region. The aH bit qualifies the address field (ADDR) to the region of a16K, a32K and a64k. If aH is set, then the ADDR field maps to that region. See ADDR description for more information.</p> <p>0 Address captured or to be accessed is not from high address region.<br/>1 Address captured or to be accessed is from high address region.</p>           |
| 2<br>aM  | <p>Address Mid region. The aM bit qualifies the address field (ADDR) to the region of a16K, a32K and a64k. If aM is set, then the ADDR field maps to that region. See ADDR description for more information.</p>                                                                                                                                                               |

*Table continues on the next page...*

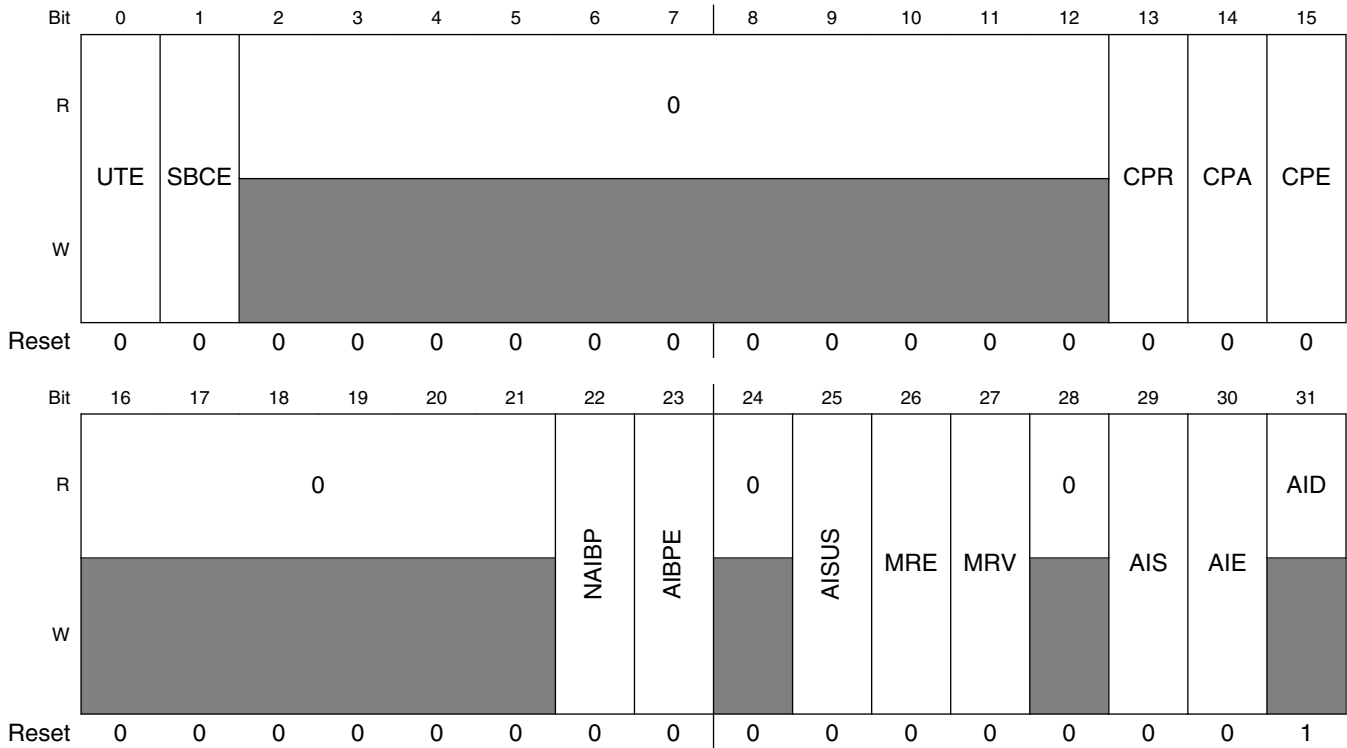
**C55FMC\_ADR field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | 0 Address captured or to be accessed is not from mid address region.<br>1 Address captured or to be accessed is from mid address region.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 3<br>aL           | Address Low region. The aL bit qualifies the address field (ADDR) to the region of a16K, a32K and a64k. If aL is set, then the ADDR field maps to that region. See ADDR description for more information.<br><br>0 Address captured or to be accessed is not from low address region.<br>1 Address captured or to be accessed is from low address region.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 4<br>a256k        | Address 256 KB region. The a256k bit qualifies the address field (ADDR) to the region. If a256k is set, then the ADDR field maps to that region. See ADDR description for more information.<br><br>The a256k register is writable.<br><br>0 Address captured or to be accessed is not from 256 KB region.<br>1 Address captured or to be accessed is from 256 KB region.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 5<br>a64k         | Address 64 KB region. The a64k bit qualifies the address field (ADDR) to the region. If a64k is set, then the ADDR field maps to that region. See ADDR description for more information.<br><br>0 Address captured or to be accessed is not from 64 KB region.<br>1 Address captured or to be accessed is from 64 KB region.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 6<br>a32k         | Address 32 KB region. The a32k bit qualifies the address field (ADDR) to the region. If a32k is set, then the ADDR field maps to that region. See ADDR description for more information.<br><br>0 Address captured or to be accessed is not from 32 KB region.<br>1 Address captured or to be accessed is from 32 KB region.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 7<br>a16k         | Address 16 KB region. The a16k bit qualifies the address field (ADDR) to the region. If a16k is set, then the ADDR field maps to that region. See ADDR description for more information.<br><br>0 Address captured or to be accessed is not from 16 KB region.<br>1 Address captured or to be accessed is from 16 KB region.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 8–28<br>ADDR      | Address. The ADR provides the first failing address in the event of ECC event error (EER set), Single Bit Correction (SBC set), as well as providing the address of a failure that may have occurred in a state machine operation (PEG cleared). ECC event errors take priority over Single Bit Corrections, which take priority over state machine errors. This is especially valuable in the event of an RWW operation, where the read senses an ECC error or Single Bit correction, and the state machine fails simultaneously. The failing address for ECC event error is held until EER is cleared. The failing address for Single Bit Correction is held until an ECC event error, or until SBC is cleared. State machine address is held until an ECC event error or Single Bit Correction event occurs, or until the next state machine event (PEG cleared) occurs. This address is always a doubleword address that selects 64 bits.<br><br>ADDR[23:3] is an offset from a base address of 0x0 for each block size region. The aH, aM, aL, a16k, a32k, a64k, and a256k qualify the block size region the ADDR field. |
| 29–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |

26.5.12 UTest 0 register (C55FMC\_UT0)

The User Test 0 (UT0) register provides a means to control UTest. UTest mode gives the ability to perform test features on the flash memory. This register is only writable when the flash memory is put into UTest mode by writing a passcode.

Address: 0h base + 54h offset = 54h



C55FMC\_UT0 field descriptions

| Field     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>UTE  | <p>U-Test Enable. This status bit gives indication when U-Test is enabled. All bits in UT0, UM0, UM1, UM2, UM3, UM4, UM5, UM6, UM7, UM8 and UM9 are locked when this bit is 0. This bit is not writeable to a 1, but may be cleared. The reset value is 0. The method to set this bit is to provide a password, and if the password matches, the UTE bit is set to reflect the status of enabled, and is enabled until it is cleared by a register write. The UTE password will only be accepted if PGM = 0 and ERS = 0 (program and erase are not being requested). UTE can only be cleared if AID = 1, MRE and AIE = 0. While clearing UTE, writes to set AIE or MRE will be ignored. For UTE, the password 0xF9F9_9999 must be written to the UT0 register.</p> <p>0 All bits in the UT0, UM0, UM1, UM2, UM3, UM4, UM5, UM6, UM7, UM8 and UM9 registers are locked<br/>1 Bits in the UT0, UM0, UM1, UM2, UM3, UM4, UM5, UM6, UM7, UM8 and UM9 registers are unlocked</p> |
| 1<br>SBCE | <p>Single Bit Correction Enable. SBCE enables Single Bit Correction results to be observed in SBC. ECC corrections that occur when SBCE is cleared will not be logged.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

Table continues on the next page...



**C55FMC\_UT0 field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | 0 Single Bit Corrections observation is disabled.<br>1 Single Bit Correction observation is enabled.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 2–12<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 13<br>CPR         | Customer Programmable Read Voltage and Reference Detection. This bit is used to control the error generation for the Customer Programmable Detection area in the UTest block. If this bit is set, and the Customer Programmable Detection location in UTest is read, a Read Voltage and Reference Error detection will be noted. This bit may not be set while CPE and CPA are set. If CPE or CPA are high, this bit will be locked. If attempts are made to write CPR, CPA or CPE simultaneously, none of them will be written.<br><br>0 Customer Programmable Read Voltage and Reference Detection Error is not enabled<br>1 Customer Programmable Read Voltage and Reference Detection Error is enabled |
| 14<br>CPA         | Customer Programmable Address Encode Detection. This bit is used to control the error generation for the Customer Programmable Detection area in the UTest block. If this bit is set, and the Customer Programmable Detection location in UTest is read, a Address Encode Error detection will be noted. This bit may not be set while CPE and CPR are set. If CPE or CPR are high, this bit will be locked. If attempts are made to write CPR, CPA or CPE simultaneously, none of them will be written.<br><br>0 Customer Programmable Address Encode Detection Error is not enabled<br>1 Customer Programmable Address Encode Detection Error is enabled                                                 |
| 15<br>CPE         | Customer Programmable EDC after ECC Detection. This bit is used to control the error generation for the Customer Programmable Detection area in the UTest block. If this bit is set, and the Customer Programmable Detection location in UTest is read, a EDC after ECC Error detection will be noted. This bit may not be set while CPR and CPA are set. If CPR or CPA are high, this bit will be locked. If attempts are made o write CPR, CPA or CPE simultaneously, none of them will be written.<br><br>0 Customer Programmable EDC after ECC Detection Error is not enabled<br>1 Customer Programmable EDC after ECC Detection Error is enabled                                                      |
| 16–21<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 22<br>NAIBP       | Next Array Integrity Break Point. If AIBPE is set, NAIBP will be set once a single bit correction (if enabled) or double bit detection is noted during the Array Integrity test. NAIBP is not writable to 1, but may be cleared to 0. Clearing NAIBP to 0, will enable the Array Integrity operation to be re-started after a breakpoint is encountered. If the Array Integrity operation completes without encountering another correction or detection, AID will be set with NAIBP remaining 0.<br><br>1 Array Integrity state machine is at a break point.<br>0 Array Integrity state machine is not currently at a break point.                                                                        |
| 23<br>AIBPE       | Array Integrity Break Point Enable. If you want to enable breakpoints during an array integrity test, AIBPE may be set. See NAIBP description for more information about array integrity breakpoints.<br><br>1 Array Integrity breakpoints are enabled during Array Integrity Checks.<br>0 Array Integrity breakpoints are not enabled.                                                                                                                                                                                                                                                                                                                                                                    |
| 24<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 25<br>AISUS       | Array Integrity Suspend. AISUS enables a suspend of an Array Integrity Operation. Array Integrity may be suspend by Setting AISUS, and resumed by clearing AISUS. AISUS is writeable only when AID is low. AISUS is clearable only when AID is high. Attempting to write AISUS and AIE on the same clock cycle will result in only AIE getting written.                                                                                                                                                                                                                                                                                                                                                    |

*Table continues on the next page...*

**C55FMC\_UT0 field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | <p>1 Array integrity sequence is suspended.</p> <p>0 Array integrity sequence not suspended.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 26<br>MRE      | <p>Margin Read Enable. MRE combined with MRV enables user margin reads to be done. Normal user reads are not affected by MRE, although user reads while the margin read operation is ongoing are not supported. MRE is not writable if AID is low, or if AISUS is high, or if NAIBP is high.</p> <p>1 Margin reads are enabled.</p> <p>0 Margin reads are not enabled.</p>                                                                                                                                                                                                                                                                                                  |
| 27<br>MRV      | <p>Margin Read Value. MRV selects the margin level that is being checked. Margin can be checked to an erased level (MRV=1) or to a programmed level (MRV=0). In order for this value to be valid, MRE must also be set. MRV is not writable if AID is low, or if AISUS is high, or if NAIBP is high.</p> <p>1 One's margin reads are requested.</p> <p>0 Zero's margin reads are requested.</p>                                                                                                                                                                                                                                                                             |
| 28<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 29<br>AIS      | <p>Array Integrity Sequence. AIS determines the address sequence to be used during array integrity checks. The default sequence (AIS = 0) is meant to replicate sequences that normal user code follows, and thoroughly checks the read propagation paths. This sequence is proprietary. The alternative sequence (AIS = 1) is logically sequential.</p> <p>It should be noted that the time to run a sequential sequence is shorter than the time to run the proprietary sequence. AIS is not writeable if AIE is high.</p> <p>1 Array integrity sequence is sequential.</p> <p>0 Array integrity sequence is proprietary sequence.</p>                                    |
| 30<br>AIE      | <p>Array Integrity Enable. AIE set to one starts the array integrity check done on all selected blocks. The address sequence is determined by AIS, and the MISR (UM0 through UM9) can be checked after the operation is complete, to determine if a correct signature has been obtained. Once an array integrity operation is requested (AIE=1), it may be terminated by clearing AIE if the operation has finished (AID = 1) or aborted by clearing AIE if the operation is ongoing (AID = 0). AIE is not simultaneously writable to a 1 as UTE is being cleared to a 0.</p> <p>0 Array integrity checks are not enabled.</p> <p>1 Array integrity checks are enabled.</p> |
| 31<br>AID      | <p>Array Integrity Done. AID is cleared upon an array integrity check being enabled (to signify the operation is ongoing). Once completed, AID is set to indicate that the array integrity check is complete. At this time the MISR (UMR registers) can be checked. AID may also assert if breakpoints are enabled (AIBPE is set), an abort is requested or a suspend is requested. AID cannot be written, and is status only.</p> <p>0 Array integrity check is ongoing.</p> <p>1 Array integrity check is done.</p>                                                                                                                                                       |

## 26.5.13 UMISR register (C55FMC\_UMn)

The Multiple Input Signature registers provide a means to evaluate array integrity.

Address: 0h base + 58h offset + (4d × i), where i=0d to 8d

|       |      |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | MISR |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |      |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0    | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

### C55FMC\_UMn field descriptions

| Field        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>MISR | <p>MISR[31:0]. The MISR registers accumulate a signature from an array integrity event. The MISR captures all data fields, as well as ECC fields, and the ECC error signal. Data (read and ECC) captured in the MISR is after the ECC logic, and represents corrected data (if required).</p> <p>The MISR can be seeded to any value by writing the MISR registers.</p> <p><b>NOTE:</b> Writing the MISR register during an array integrity operation (including suspend or breakpoint) although possible, is not recommended. A write of the MISR registers at this point may alter the final signature in an unpredictable way.</p> <p>The MISR register provides a means to calculate a MISR during array integrity operations.</p> <p>The MISR can be represented by the following polynomial:</p> $x^{289} + x^7 + x^6 + x^5 + x^4 + x^2 + 1$ <p>The MISR is calculated by taking the previous MISR value and then "exclusive ORing" the new data. In addition the most significant bit (in this case it is MISR[288]), is then "exclusive ORed" into input of MISR[7], MISR[6], MISR[5], MISR[4], MISR[2] and MISR[0]. The result of the "exclusive OR" is shifted left on each read.</p> <p>The MISR register is used in array integrity operations.</p> <p>If during address sequencing, reads extend into an invalid address location (in other words, greater than the maximum address for a given array size) then reads are still executed to the array, but the results from the array read are not deterministic. In this instance, the MISR registers is not recalculated, and the previous value is retained.</p> |

## 26.5.14 UMISR register (C55FMC\_UM9)

The Multiple Input Signature registers provide a means to evaluate array integrity.

Address: 0h base + 7Ch offset = 7Ch

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31   |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    | MISR |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    |

### C55FMC\_UM9 field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–30<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 31<br>MISR       | <p>MISR[288].</p> <p>The MISR registers accumulate a signature from an array integrity event. The MISR captures all data fields, as well as ECC fields, and the ECC error signal. Data (read and ECC) captured in the MISR is after the ECC logic, and represents corrected data (if required).</p> <p>The MISR can be seeded to any value by writing the MISR registers.</p> <p><b>NOTE:</b> Writing the MISR register during an array integrity operation (including suspend or breakpoint) although possible, is not recommended. A write of the MISR registers at this point may alter the final signature in an unpredictable way.</p> <p>The MISR register provides a means to calculate a MISR during array integrity operations.</p> <p>The MISR can be represented by the following polynomial:</p> $x^{289} + x^7 + x^6 + x^5 + x^4 + x^2 + 1$ <p>The MISR is calculated by taking the previous MISR value and then "exclusive ORing" the new data. In addition the most significant bit (in this case it is MISR[288]), is then "exclusive ORed" into input of MISR[7], MISR[6], MISR[5], MISR[4], MISR[2] and MISR[0]. The result of the "exclusive OR" is shifted left on each read.</p> <p>The MISR register is used in array integrity operations.</p> <p>If during address sequencing, reads extend into an invalid address location (in other words, greater than the maximum address for a given array size) then reads are still executed to the array, but the results from the array read are not deterministic. In this instance, the MISR registers is not recalculated, and the previous value is retained</p> |

### 26.5.15 Over-Program Protection 0 register (C55FMC\_OPP0)

The Over-Program Protection 0 (OPP0) register provides a means to protect blocks from being over-programmed. This register shows the over-program protection status.

#### NOTE

The reset value of this register is set at the factory and varies among devices. See the chip-specific C55FMC information for details.

Address: 0h base + 80h offset = 80h

| Bit   | 0 | 1      | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16     | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|-------|---|--------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| R     | 0 | LOWOPP |    |    |    |    |    |    |    |    |    |    |    |    |    |    | MIDOPP |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0      | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0*     | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* |    |

\* Notes:

- MIDOPP field: The reset value of the MIDOPP field is set at the factory and varies among devices. See the chip-specific C55FMC information for details.
- LOWOPP field: The reset value of the LOWOPP field is set at the factory and varies among devices. See the chip-specific C55FMC information for details.

#### C55FMC\_OPP0 field descriptions

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–1<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 2–15<br>LOWOPP  | Low Block Over-Program Protection[13:0].<br><br>A value of 1 in a bit of the over-program protection register signifies that the corresponding block is protected from over-programming. A value of 0 in the OPP register signifies that the corresponding block is available to be over-programmed. The block numbering for the Low Blocks starts at LOWOPP[0] with 16 KB block region (if available), then the 32 KB block region (if available), and lastly the 64 KB block region (if available). This continues until all blocks are accounted for.<br><br>The LOWOPP register is not writable, and is status only.<br><br>The default value of the LOWOPP bits is not OPP. In the event that blocks are not present (due to configuration or total memory size), the OPP bits default not OPP, and will remain 0.<br><br>1 The corresponding block is protected from over-programming.<br>0 The corresponding block is available to be over-programmed. |
| 16–31<br>MIDOPP | Mid Block Over-Program Protection[15:0].<br><br>A value of 1 in a bit of the over-program protection register signifies that the corresponding block is protected from over-programming. A value of 0 in the OPP register signifies that the corresponding block is available to be over-programmed.<br><br>The block numbering for the Mid Blocks starts at MIDOPP[0] with 16 KB block region (if available), then the 32 KB block region (if available), and lastly the 64 KB block region (if available). This continues until all blocks are accounted for.                                                                                                                                                                                                                                                                                                                                                                                               |

Table continues on the next page...

**C55FMC\_OPP0 field descriptions (continued)**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                            |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | <p>The MIDOPP register is not writable, and is status only.</p> <p>The default value of the MIDOPP bits is not OPP. In the event that blocks are not present (due to configuration or total memory size), the OPP bits default not OPP, and will remain 0.</p> <p>1 The corresponding block is protected from over-programming.<br/> 0 The corresponding block is available to be over-programmed.</p> |

**26.5.16 Over-Program Protection 1 register (C55FMC\_OPP1)**

The Over-Program Protection 1 (OPP1) register provides a means to protect blocks from being over programmed. This register shows the over-program protection status.

**NOTE**

The reset value of this register is set at the factory and varies among devices. See the chip-specific C55FMC information for details.

Address: 0h base + 84h offset = 84h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|---------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16      | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | HIGHOPP |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0*      | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* |    |

\* Notes:

- HIGHOPP field: The reset value of the HIGHOPP field is set at the factory and varies among devices. See the chip-specific C55FMC information for details.

**C55FMC\_OPP1 field descriptions**

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 16–31<br>HIGHOPP | <p>High Block Over-Program Protection[15:0].</p> <p>A value of 1 in a bit of the over-program protection register signifies that the corresponding block is protected from over-programming. A value of 0 in the OPP register signifies that the corresponding block is available to be over-programmed. The block numbering for the High Blocks starts at HIGHOPP[0] with 16 KB block region (if available), then the 32 KB block region (if available), and lastly the 64 KB block region (if available). This continues until all blocks are accounted.</p> <p>The HIGHOPP register is not writable, and is status only.</p> <p>The default value of the HIGHOPP bits is not OPP. In the event that blocks are not present (due to configuration or total memory size), the OPP bits default not OPP, and will remain 0.</p> <p>1 The corresponding block is protected from over-programming.<br/> 0 The corresponding block is available to be over-programmed.</p> |

### 26.5.17 Over-Program Protection 2 register (C55FMC\_OPP2)

The Over-Program Protection 2 (OPP2) register provides a means to protect blocks from being over programmed. This register shows the over-program protection status.

#### NOTE

The reset value of this register is set at the factory and varies among devices. See the chip-specific C55FMC information for details.

Address: 0h base + 88h offset = 88h

|       |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0        | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | A256KOPP |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0*       | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* |

\* Notes:

- A256KOPP field: The reset value of the A256KOPP field is set at the factory and varies among devices. See the chip-specific C55FMC information for details.

#### C55FMC\_OPP2 field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>A256KOPP | <p>256K Block Over-Program Protection[31:0].</p> <p>A value of 1 in a bit of the over-program protection register signifies that the corresponding block is protected from over-programming. A value of 0 in the OPP register signifies that the corresponding block is available to be over-programmed. The block numbering for the 256 KB Blocks starts at A256KOPP[0] and continues until all blocks are accounted.</p> <p>The 256KOPP register is not writable, and is status only.</p> <p>The default value of the A256KOPP bits is not OPP. In the event that blocks are not present (due to configuration or total memory size), the OPP bits default not OPP, and will remain 0.</p> <p>1 The corresponding block is protected from over-programming.<br/>0 The corresponding block is available to be over-programmed.</p> |

### 26.5.18 Over-Program Protection 3 register (C55FMC\_OPP3)

The Over-Program Protection 3 (OPP3) register provides a means to protect blocks from being over programmed. This register shows the over-program protection status.

#### NOTE

The reset value of this register is set at the factory and varies among devices. See the chip-specific C55FMC information for details.

## Memory map and register definition

Address: 0h base + 8Ch offset = 8Ch

| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16       | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | A256KOPP |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0*       | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* |    |

\* Notes:

- A256KOPP field: The reset value of the A256KOPP field is set at the factory and varies among devices. See the chip-specific C55FMC information for details.

## C55FMC\_OPP3 field descriptions

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 16–31<br>A256KOPP | 256K Block Over-Program Protection[47:32].<br><br>A value of 1 in a bit of the over-program protection register signifies that the corresponding block is protected from over-programming. A value of 0 in the OPP register signifies that the corresponding block is available to be over-programmed. The block numbering for the 256 KB Blocks starts at 256KOPP[0] and continues until all blocks are accounted.<br><br>The 256KOPP field is not writable, and is status only.<br><br>The default value of the A256KOPP bits is not OPP. In the event that blocks are not present (due to configuration or total memory size), the OPP bits default not OPP, and will remain 0.<br><br>1 The corresponding block is protected from over-programming.<br>0 The corresponding block is available to be over-programmed. |

## 26.5.19 Test Mode Disable Password Check register (C55FMC\_TMD)

The Test Mode Disable Password Check (TMD) register provides a means to provide a challenge password to disable the Test Mode Disable Seal block select.

Writes to the register have no effect except for a password challenge. Reads to this register always return 0.

Address: 0h base + 90h offset = 90h

| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16  | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | PWD |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

## C55FMC\_TMD field descriptions

| Field       | Description        |
|-------------|--------------------|
| 0–31<br>PWD | Password challenge |



## 26.6 Functional Description

The embedded flash memory consists of address spaces in four groupings:

- Low address space
- Mid address space
- High address space
- 256 KB address space

The main address space is divided into partitions. Each address space mentioned above consists of a partition pair. Partitions are used to determine locations for valid read-while-write (RWW) operations. While the embedded flash memory is performing a write (program or erase) to a given partition, it can simultaneously perform a read from any other partition. For program operations, only the address specified by an interlock write determines the partition being written (block locking and block select registers do not determine the RWW partitions being written). For erase operations, only blocks that are selected and unlocked determine the RWW partitions being written.

The main address space is also divided into blocks to implement independent erase and program protection. The UTest NVM block also exists as a block, and has independent program protection. The UTest NVM block is included to support systems that require nonvolatile memory (NVM) for security or to store system initialization information.

A number of MCR bits are protected against write when another bit, or set of bits, is in a specific state. These write locks are covered on a bit-by-bit basis in [Module Configuration Register \(C55FMC\\_MCR\)](#). The write locks do not consider the effects of trying to write two or more bits simultaneously. The embedded flash memory does not allow bits to be written simultaneously which put the device into an illegal state. This is implemented through a priority mechanism among the bits. The bit changing priorities are detailed in the following table.

**Table 26-32. MCR bit set/clear priority levels**

| Priority level | MCR Bit(s) |
|----------------|------------|
| 1              | ERS        |
| 2              | PGM        |
| 3              | EHV        |
| 4              | ESUS, PSUS |

If two or more MCR bits are written simultaneously only the bit with the lowest priority number is accepted for modification. Setting two bits with the same priority number is prevented by existing write locks or do not put the flash memory in an illegal state.

For example, setting ERS and PGM simultaneously results in only ERS being set. Attempting to clear EHV while setting PSUS results in EHV being cleared, while PSUS is unaffected.

Each read of the embedded flash memory retrieves a page, or eight consecutive words (256 bits), of information. The address for each word retrieved within a page differs from the other addresses in the page only by address bits [4:2]. The flash memory page read architecture easily supports both cache and burst mode at the BIU level for high-speed read applications.

The embedded flash memory supports fault tolerance through error correction code (ECC) and error detection. ECC implemented within the embedded flash memory corrects single-bit failures and detects double-bit failures.

A software mechanism is provided to independently lock and unlock each block.

Program and erase of the embedded flash memory requires multiple system clock cycles to complete. The program and erase sequence may be suspended or aborted.

The embedded flash memory may reside in various modes. The modes that are available include:

- User mode
- UTest mode

Each of these modes is discussed in detail in the following sections.

## **26.6.1 User Mode**

In user mode the embedded flash memory may be read and written (register writes and interlock writes), programmed or erased. The following sub-sections define all actions that may be performed in user mode.

### **26.6.1.1 Read and write**

The default state of the embedded flash memory is read. The main and UTest NVM address space can be read only in the read state. The MCR is always available for read. The embedded flash memory enters the read state on reset. The embedded flash memory is in the read state under four sets of conditions:

- The read state is active when the embedded flash memory is enabled (User Mode Read).
- The read state is active when MCR[PGM] or MCR[ERS] are high and high-voltage operation is ongoing (read-while-write).

### NOTE

Reads done to the partition(s) being operated on (either erased or programmed) result in an error and the MCR[RWE] bit is set.

- The read state is active when MCR[PGM] and MCR[PSUS] are high (program suspend).
- The read state is active when MCR[ERS] and MCR[ESUS] are high and MCR[PGM] is low (erase suspend).

In embedded flash memory, flash core reads return 256 bits (1 page). Register reads return 32 bits of data.

### NOTE

Flash core reads are done through the BIU. In many cases the BIU does read page buffering to allow sequential reads to be done with higher performance. This could provide a data coherency issue that must be handled with software. Data coherency may be an issue after a program or erase operation, as well as UTest NVM block operations.

In user mode, registers may be written. Array may be written to do interlock writes.

Register reads to unmapped register address space return all zeroes.

Register writes to unmapped register address space have no effect.

Interlock writes that are attempted during a high-voltage operation (MCR[EHV] = 1 or MCR[DONE] = 0) will result in the interlock write being ignored, and address and data will not be updated.

### NOTE

Care must be taken when doing 32-bit interlock writes. Performing two 32-bit interlock writes to the same word location is not supported. In addition, mixing 32-bit interlock writes with 64-bit interlock writes to the same double-word location is not supported. If these combinations are done, and a program operation is executed, the result will be MCR[PEG] = 0. See the Module Configuration Register (MCR) section for more information.

## 26.6.1.2 Program

A flash memory program sequence operates on any page within the flash core. Within a page, up to eight words may be altered in a single program operation. Also, up to four pages can be altered in a single program operation. Whenever the array is programmed, the ECC bits also get programmed. ECC is handled on a 64-bit boundary. Thus, if only one word in any given 64-bit ECC segment is programmed, the adjoining word (in that segment) should not be programmed, since ECC calculation has already completed for that 64-bit segment. Attempts to program the adjoining word will probably result in an operation failure. It is recommended that all programming operations be from 64 bits to 1024 bits, and be 64-bit aligned. The programming operation should completely fill selected ECC segments within the page. Only one program is allowed per 64-bit ECC segment between erases.

### Warning

In rare cases over-programming of a 64-bit ECC segment may be done (EEPROM emulation).

Programming changes the value stored in an array bit from logic 1 to logic 0 only. Programming cannot change a stored logic 0 to a logic 1.

### NOTE

If a logic 0 is attempted to be over-programmed by a logic 1, the resulting operation will fail ( $\text{MCR}[\text{PEG}] = 0$ ), and the 0's that are interlocked will be merged (ORed) with 0's that are already present in the 64-bit ECC segment, unless the block is designated as an Over-Program Protected block.

Addresses in locked/disabled blocks cannot be programmed. Values may be programmed in any or all of eight words, within a page, with a single program sequence. Page-bound words have addresses which differ only in address bits [4:2]. Up to four pages can be programmed at once on a quad-page boundary, which differ only in address bits [6:5]. The program operation consists of the following sequence of events:

1. Change the value in the  $\text{MCR}[\text{PGM}]$  bit from a 0 to a 1.

### NOTE

Ensure the block that contains the address to be programmed is unlocked.

2. Write the first address to be programmed with the program data. The embedded flash memory latches address bits [22:7] and SoC-specific UTest NVM enable at this time. The embedded flash memory latches data written as well. This write is referred to as

a program data interlock write. An interlock write may be done as a 64-bit transaction or a 32-bit transaction. Refer to the Flash Memory Controller chapter for information on the size of writes that are allowed.

3. If more than one word or doubleword is to be programmed, write each additional address in the page with data to be programmed. This is referred to as a program data write. The embedded flash memory ignores address bits [22:7] and SoC-specific UTest NVM enable for program data writes. All unwritten data words default to 0xFFFF\_FFFF.
4. Write a logic 1 to the MCR[EHV] bit to start the internal program sequence or skip to step 9 to terminate.
5. Wait until the MCR[DONE] bit goes high.

### NOTE

Since MCR[DONE] clears with MCR[EHV] being set, it may not be possible for software to read MCR[DONE] as a 0 prior to this step, depending on the operation selected.

6. Confirm MCR[PEG] = 1.
7. Write a logic 0 to the MCR[EHV] bit.
8. If more addresses are to be programmed, return to step 2.
9. Write a logic 0 to the MCR[PGM] bit to terminate the program sequence.

The first write after a program is initiated determines the quad-page address to be programmed. Program may be initiated with the 0 to 1 transition of the MCR[PGM] bit or by clearing the MCR[EHV] bit at the end of a previous program. This first write is referred to as an interlock write. The interlock write determines if the UTest NVM or normal array space is to be programmed by sampling SoC-specific UTest NVM enable and causing MCR[PEAS] to be set/cleared.

In the case of an erase-suspended program, the values in MCR[PEAS] may be modified via the program interlock write, enabling erase-suspended programs to and from UTest NVM space.

An interlock write must be performed before setting MCR[EHV]. A program sequence may be terminated by clearing MCR[PGM] prior to setting MCR[EHV].

After the interlock write, additional writes affect the data to be programmed at the word location determined by address bits [4:2], as well as the page location within a 1024-bit segment (determined by address bits [6:5]). Unwritten locations default to a data value of 0xFFFF\_FFFF. If multiple writes are done to the same location, the data for the last write is used in programming.

While DONE is low, EHV is high, and PSUS is low, then EHV may be cleared, resulting in a program abort. A program abort forces the embedded flash memory to step 8 of the program sequence. An aborted program results in PEG being set low, indicating a failed operation. The data space being operated on before the abort contains indeterminate data. A program sequence may not be aborted while in program suspend.

### **Warning**

Aborting a program operation leaves the flash core addresses being programmed in an indeterminate data state. This may be recovered by executing an erase on the affected blocks.

#### **26.6.1.2.1 Program software locking**

A software mechanism is provided to independently lock/unlock blocks against program and erase.

Software locking is done through the LOCK0, LOCK1, LOCK2, and LOCK3 registers. These may be written through register writes, and may be read through register reads.

#### **26.6.1.2.2 Program hardware locking**

Hardware locking which only affects the program operation is also available for UTest, 16 KB, 32 KB, 64 KB, and 256 KB blocks.

#### **26.6.1.2.3 Program suspend/resume**

The program sequence may be suspended to allow read access to the flash core. It is not possible to erase during a program suspend, or to program during a program suspend. These operations are prevented by register locking described in the Module Configuration Register (MCR) section. Read-while-write operation may also be used to read the array during a program sequence, providing the read is to a different partition.

A program suspend can be initiated by changing the value of the MCR[PSUS] bit from a 0 to a 1. MCR[PSUS] can be set high at any time when MCR[PGM] and MCR[EHV] are high. A 0 to 1 transition of MCR[PSUS] causes the embedded flash memory to start the sequence to enter program suspend, which is a read state. MCR[DONE] must become a logic level 1 before the embedded flash memory is suspended. At this time flash core reads may be attempted. Once suspended, the flash core may only be read. During suspend, all reads to flash core locations targeted for program, and blocks targeted for erase, return indeterminate data.

The program sequence is resumed by writing a logic 0 to MCR[PSUS]. MCR[EHV] must be set to a 1 before clearing MCR[PSUS] to resume operation. When the operation resumes, the embedded flash memory continues the program sequence from one of a set of predefined points. This may extend the time required for the program operation.

#### NOTE

Repeated suspends at a high frequency may result in the operation not completing normally. Although suspend frequency is not limited, enabling at least 20  $\mu$ s between suspend resume and future suspend requests improve the opportunity for the flash to complete the operation successfully.

#### 26.6.1.2.4 Over-program protection enable

One-time programming can be enabled on a block basis. In an over-program protection enabled block, any doubleword which has already been programmed cannot be programmed again. The over-program protection enable does not affect erase operation. Attempts to over-program will result in MCR[PEG] being cleared. If any doubleword within the quad-page has an over-program protection violation, MCR[PEG] will be cleared, and no doublewords will be programmed.

One-time programming can be enabled for 16 KB, 32 KB, 64 KB, or 256 KB blocks.

Over-program protection is enabled by sideband signals, and if enabled, the program operation will be modified to check for programmed bits prior to doing a high-voltage program event.

#### 26.6.1.2.5 Successful program address

To enable the SoC to create logic to authenticate operations with a program (such as in a diary operation), the flash module provides the last successful address programmed as a sideband signal. This value will be held until the next successful program. A successful program is defined as a program operation of data that is not all 1's to an unlocked block, and the operation results in MCR[PEG] returning a 1. If any doubleword within the pages being programmed is interlocked with all 1's, the sideband signal will not be updated. An address of all 1's, and each of the block size selects selected, is the default (and invalid) address after reset.

### 26.6.1.3 Erase

Erase changes the value stored in all bits of the selected block(s) to logic 1. An erase sequence operates on any combination of blocks in the main address space. The erase sequence is fully automated within the flash memory. Blocks to be erased must be selected prior to initiating the erase sequence. Locked/disabled blocks cannot be erased. If multiple blocks are selected for erase during an erase sequence, the blocks are erased sequentially starting with the lowest numbered block and terminating with the highest (low first, mid second, high last, 16 KB first, 32 KB second, 64 KB third, last blocks are 256 KB). The erase sequence consists of the following events:

1. Change the value in the MCR[ERS] bit from 0 to 1.
2. Select the block or blocks to be erased by writing ones to the appropriate registers in SEL0, SEL1, SEL2, or SEL3 registers.

#### NOTE

Lock and Select are independent. If a block is selected and locked, no erase occurs.

3. Write to any address in flash memory. This is referred to as an erase interlock write. An erase interlock write to UTest NVM space is not allowed if sealed (since erase is not allowed in the UTest NVM space when sealed).
4. Write a logic 1 to the MCR[EHV] bit to start an internal erase sequence or skip to step 9 to terminate.
5. Wait until the MCR[DONE] bit goes high.

#### NOTE

Since MCR[DONE] clears with MCR[EHV] being set, it may not be possible for software to read MCR[DONE] as a 0 prior to this step, depending on the operation selected.

6. Confirm MCR[PEG] = 1.
7. Write a logic 0 to the MCR[EHV] bit.
8. If more blocks are to be erased, return to step 2.
9. Write a logic 0 to the MCR[ERS] bit to terminate the erase.

After setting ERS, one write, referred to as an interlock write, must be performed before EHV can be set to a 1. Data words written during erase sequence interlock writes are ignored. The erase sequence may be terminated by clearing ERS before setting EHV.

An erase operation may be aborted by clearing EHV, assuming DONE is low, EHV is high, and ESUS is low. An erase abort forces the embedded flash memory to step eight of the erase sequence. An aborted erase results in PEG being set low, indicating a failed operation. The block(s) being operated on before the abort contain indeterminate data. An erase sequence may not be aborted while in erase suspend.



### 26.6.1.3.1 Erase software locking

Software Locking affect erase operation. For details on this see [Program software locking](#).

### 26.6.1.3.2 Erase hardware locking

Hardware locking which only affects the erase operation is also available for UTest, 16 KB, 32 KB, 64 KB, and 256 KB blocks.

### 26.6.1.3.3 Erase suspend/resume

The erase sequence may be suspended to allow read access to the flash core. The erase sequence may also be suspended to program (Erase-Suspended Program) the flash core. A program started during erase suspend can in turn be suspended. Only one erase suspend and one program suspend are allowed at a time during an operation. It is not possible to erase during an erase suspend, or program during a program suspend. During suspend, all reads to flash core locations targeted for program and blocks targeted for erase return indeterminate data.

Read-while-write operation may also be used to read the array during an erase sequence, provided the read is to a partition not selected for erase.

An erase suspend can be initiated by changing the value of the MCR[ESUS] bit from a 0 to a 1. MCR[ESUS] can be set to a 1 at any time when MCR[ERS] and MCR[EHV] are high and MCR[PGM] is low. A 0 to 1 transition of MCR[ESUS] causes the embedded flash memory to start the sequence which places it in erase suspend. MCR[DONE] must become a logic level 1 before the embedded flash memory is suspended and further actions are attempted. Once suspended, the array may be read or a program sequence may be initiated (erase-suspended program). Before initiating a program sequence, MCR[EHV] must first be cleared. If a program sequence is initiated, the values of SoC-specific UTest NVM enable is recaptured. Once the erase-suspended program is completed, the value of PEAS is returned to its erase value. Flash core reads that occur while MCR[ESUS] = 1 from the block(s) being erased will return indeterminate data.

The erase sequence is resumed by writing a logic 0 to MCR[ESUS]. MCR[EHV] must be set to a 1 and MCR[PGM] must be cleared (in the event of an erase-suspended program) before MCR[ESUS] can be cleared to resume the operation. The embedded flash memory continues the erase sequence from one of a set of predefined points. This may extend the time required for the erase operation.

**NOTE**

1. In an erase-suspended program, programming flash core locations in blocks which were being operated on will respond by completing the operation with a fail code (MCR[PEG] = 0). Programming voltages will not be applied to the memory array in this case.
2. Repeated suspends at a high frequency may result in the operation not completing normally. Although suspend frequency is not limited, enabling at least 5ms between suspend resume and future suspend requests improve the opportunity for the flash to complete the operation successfully.

**26.6.2 UTest mode**

UTest mode is a mode into which customers can put the embedded flash memory so as to do specific tests that check the integrity of the embedded flash memory.

**26.6.2.1 Array integrity self check**

The Array Integrity (AI) operation is a flash-specific self-test feature in which a MISR signature value is generated from data read from selected flash blocks. After the operation is complete, customer code compares the resulting MISR signature to the expected value to determine if an incorrect read or ECC detection occurred.

Array integrity operations are fully contained within the flash module. Results of Array integrity operations are not reported beyond the boundary of the flash module.

During array integrity test, user mode array reads are ignored to ensure the array integrity operation is not corrupted. User mode array reads may be executed if suspended or at a breakpoint.

AI read operations are executed using a predefined address sequence selected via the UT0[AIS] register field value. The data to be read is customer-specific, e.g., user program code that has been programmed into the flash memory. Any random or nonrandom code is valid. The expected MISR signature is calculated based on the specific data to be read.

Note that array integrity MISR signature calculations are performed after ECC detection and correction so correctable ECC events do not cause the MISR signature to deviate from the expected value.

The Array integrity check consists of the following sequence:

1. Enable UTest mode by setting the UT0[UTE] bit to 1.
2. Select one or more blocks for an array integrity check by writing 1's to the appropriate bits in the SEL0 – SEL3 registers. Blocks selected for AI test do not need to be unlocked.

Flash blocks that have not been selected are still read as part of the array integrity sequence to enable full transition coverage. Reads on unselected blocks are not captured in the MISR. Selecting fewer blocks may not result in a faster array integrity execution time, depending on the combinations of blocks selected and the sequence.

Blocks protected with the Test Mode Disable Seal are still read as part of the array integrity sequence. The resulting read on sealed blocks is not captured in the MISR, but the Single Bit Correction, Double Bit Detection and breakpoints are still honored.

### NOTE

It is not possible to do array integrity operations on the UTest NVM block.

3. If desired, set the UT0[AIS] bit to 1 for sequential addressing only.

For normal integrity checks of the flash memory, sequential addressing is recommended. If it is required to more fully check the read path (in a diagnostic mode), it is recommended that UT0[AIS] be left at 0, to use the address sequence that checks the read path more fully, and examine read transitions. This sequence takes more time.

4. Seed the MISR registers (UM0 — UM9) with desired values.
5. If breakpoints are desired, set UT0[AIBPE] to 1, and ensure that MCR[EER] and MCR[SBC] are cleared. If it is desired to break on single-bit correction, the UT0[SBCE] must be set.
6. Initiate AI by setting the UT0[AIE] bit to 1.

The array integrity operation may be aborted prior to UT0[AID] going high. This may be done by clearing the UT0[AIE] bit and continuing to the next step. It should be noted that in the event of an aborted array integrity check the MISR registers contain a signature for the portion of the operation that was completed prior to the abort, and are not deterministic. Prior to doing another array integrity operation, the UM0 – UM9 registers may need to be initialized to the desired seed value by doing register writes.

The array integrity operation may be suspended prior to UT0[AID] going high. UT0[AISUS] may be set to request an array integrity suspend. After the UT0[AISUS] bit is set, the user should wait for UT0[AID] bit to go high, which

indicates the flash has entered the suspend state, and normal reads to the flash may be done. After the UT0[AID] bit goes high, the UT0[AISUS] bit may be cleared to resume the array integrity sequence.

7. Wait until the UT0[AID] bit goes high, indicating that array integrity test is complete.
8. If breakpoints are enabled, check the UT0[NAIBP] bit. If set to 1, the MCR[EER], MCR[SBC], and ADDR registers may be checked to determine the cause of the break and the address of the break. Prior to resuming the operation, clear the MCR[SBC] or MCR[EER] bits, then resume the operation by clearing the UT0[NAIBP] bit. Continue to wait until the UT0[AID] bit goes high. If breakpoints are not enabled, or if the UT0[NAIBP] bit is low when the UT0[AID] bit goes high, the operation is complete. Continue to the next step.
9. Read values in the MISR registers (UM0 — UM9) to ensure correct signature.

### NOTE

Array integrity reads may be done to unselected (or non-present) locations. Reads done to these locations do not update the MISR, and do not update the MCR[EER] and MCR[SBC] error bits.

10. Write a logic 0 to the UT0[AIE] bit to complete array integrity test.

## 26.6.2.2 User margin read

User Margin Read is a flash-specific self-test feature that enables reads to be done at a Program margin or Erase margin level.

Margin reads are performed with an internal read reference voltage that has been adjusted, i.e., "margined" toward the threshold voltage being checked. The read reference voltage is either margined toward a 0 (programmed bit) or margined towards a 1 (erased bit), depending on the margin read operation selected.

User margin read is done using the array integrity interface, and has all the associated features of the array integrity interface (MISR and Breakpoints). User margin read is a self-timed event, and is independent of system clocks or wait states selected.

Margin read operations are fully contained within the flash module. Results of margin read operations are not reported beyond the boundary of the flash module.

During margin read operations, user mode array reads are ignored to ensure the operation is not corrupted. User mode array reads may be executed if suspended or at a breakpoint.

ECC corrections and detections are noted during the user margin read test. Note that margin read MISR signature calculations are performed *after* ECC detection and correction so correctable ECC events do not cause the MISR signature to deviate from the expected value.

The data to be read is customer-specific, e.g., user program code that has been programmed into the flash memory. Any random or non-random code is valid. The expected MISR signature is calculated based on the specific data to be read. After the operation is completed the margin read results can be checked by reading the MCR[EER] and MCR[SBC] bits to determine if zero, one, or two bits are being detected by the margin read, and comparing the generated MISR signature to the expected value.

The use model for margin read is in the event of a user-detected single bit correction (through user reads). A margin read may be done to check for a possible second bit falling within the selected margin levels.

The procedure for doing margin reads is:

1. Enable UTest mode by setting the UT0[UTE] bit to 1.
2. Select one or more blocks for margin read check by writing 1's to the appropriate bits in SEL0 – SEL3 registers. Blocks selected for margin read do not need to be unlocked.

Flash blocks that have not been selected are still read as part of the margin read sequence to enable full transition coverage. Reads on unselected blocks are not captured in the MISR. Selecting fewer blocks does not result in a faster margin read execution time.

Blocks protected with the Test Mode Disable Seal are still read as part of the margin read sequence. The resulting read on sealed blocks is not captured in the MISR, but the Single Bit Correction, Double Bit Detection and breakpoints are still honored.

### NOTE

It is not possible to do margin read operations on the UTest NVM block.

3. Set the UT0[AIS] bit to 1 for sequential addressing only.
4. Seed the MISR registers (UM0 — UM9) with desired values.
5. If breakpoints are desired, set UT0[AIBPE] to 1, and ensure that MCR[EER] and MCR[SBC] are cleared. If it is desired to break on single-bit correction, the UT0[SBCE] bit must be set to 1.
6. Initiate margin read by setting the UT0[MRE] bit to 1.
7. Select either one's margin or zero's margin by setting the UT0[MRV] bit to the appropriate value.
8. Set the UT0[AIE] bit.

During margin read operations, with UT0[AID] low, it is not recommended to attempt write operations to the MCR[SBC] and MCR[EER] bits. It is recommended that these bits only be written during suspend, breakpoints, or at completion of the Margin Read operation.

9. Wait until the UT0[AID] bit goes high, indicating that margin read test is complete.

If breakpoints were enabled or a suspend was requested during margin read, the operation may be at a breakpoint or a suspend state. See Step 8 in section [Array integrity self check](#) for more information.

10. Read values in the MISR registers (UM0 — UM9) to ensure correct signature.

#### NOTE

Margin reads may be done to unselected (or non-present) locations. Reads done to these locations do not update the MISR, and do not update the MCR[EER] and MCR[SBC] error bits.

11. Write a logic 0 to the UT0[AIE] bit.

## 26.7 Initialization information

A reset is the highest priority operation for the embedded flash memory and terminates all other operations.

The embedded flash memory uses reset to initialize register and status bits to their default reset values. If the embedded flash memory is executing a program or erase operation (PGM or ERS = 1) and a reset is issued, the operation is aborted and the embedded flash memory disables the high-voltage logic without damage to the high-voltage circuits. Reset aborts all operations and forces the embedded flash memory into user mode ready to receive accesses.

After reset is requested, MCR[DONE] goes low, and remains low during reset and reset recovery. At the end of reset recovery, MCR[DONE] transitions from a 0 to a 1.

After reset is completed, register reads may be done, although it should be noted that registers that require updating from UTest NVM information, or other inputs, may not read updated values until MCR[DONE] transitions high.

During reset recovery, register writes are not allowed until the MCR[DONE] bit transitions high to indicate reset recovery is completed.

### **Warning**

Resetting during a program or erase operation leaves the flash core blocks being programmed or erased in an indeterminate data state. This may be recovered by executing an erase on the affected blocks.





## Chapter 27

# Analog Modules Overview

### 27.1 Introduction

This chip has the following modules for the primary purpose of analog measurement:

- Two EQADC modules, each containing two internal ADCs
- Four Sigma-Delta (SD) ADC modules
- Two Temperature Sensors

### 27.2 EQADC subsystem

The integration of the two EQADC modules on this chip, EQADC\_A and EQADC\_B, is orthogonal except for the connection of the ADCs to specific external pins and internal channels.

The following diagram shows the connections of the two EQADCs to other on-chip components and external pins.

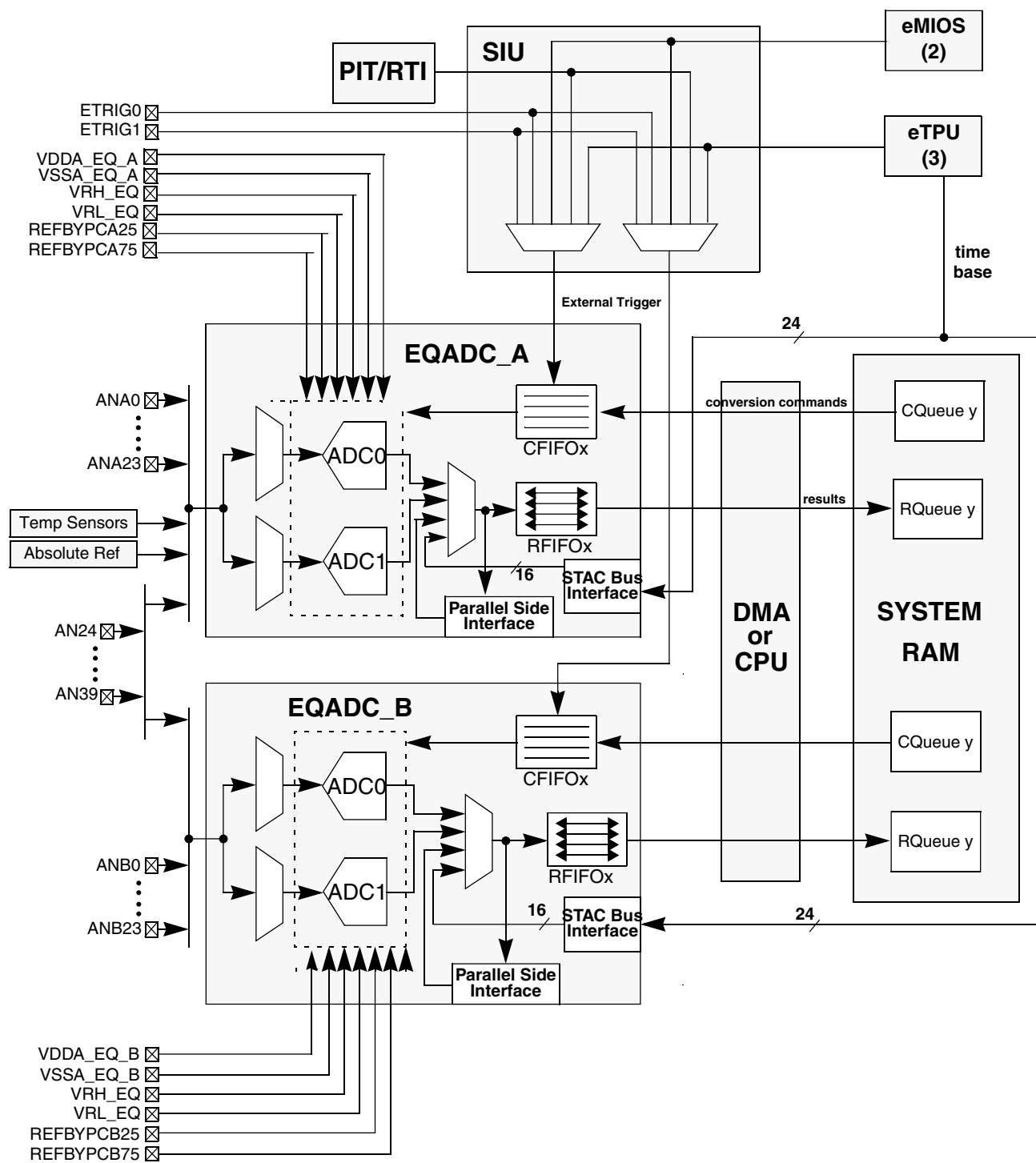


Figure 27-1. EQADC subsystem

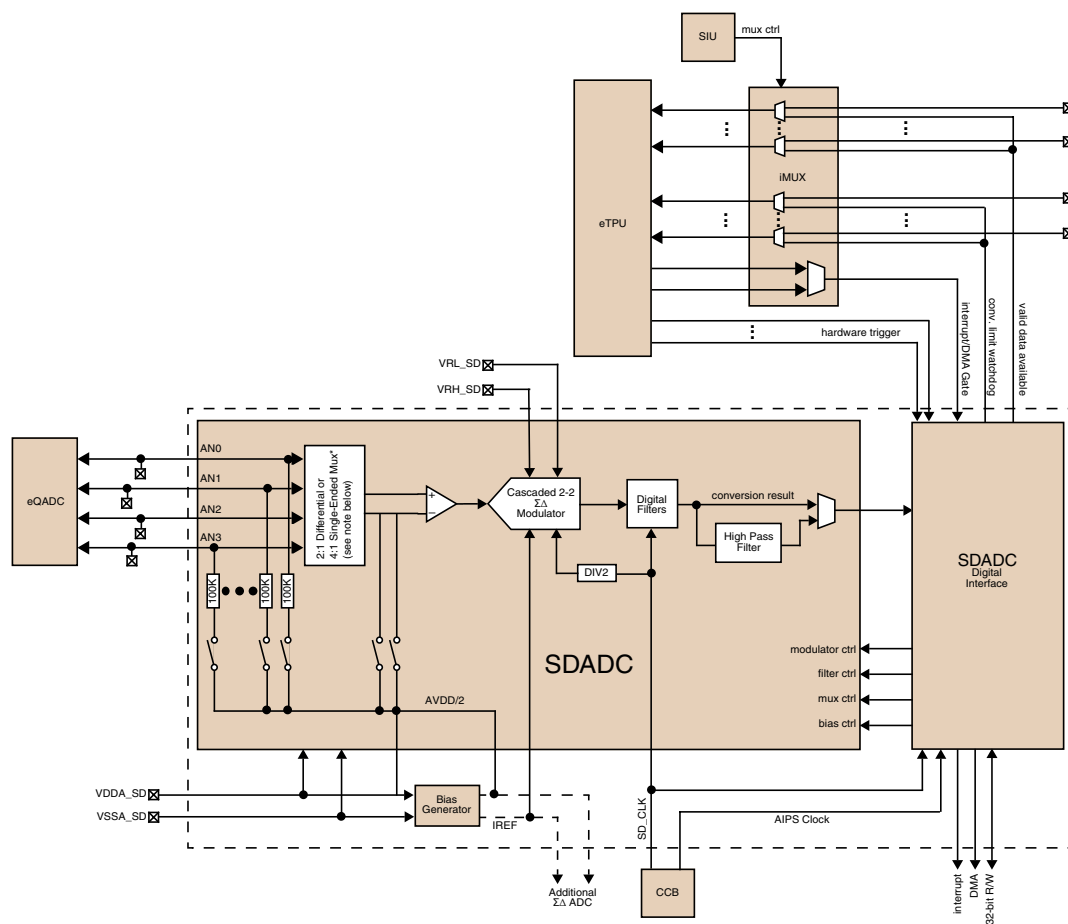
## 27.3 SDADC connections to other modules

The integration of the four SDADC modules on this chip is orthogonal except for the connection of the ADCs to specific external pins and internal channels.

Each SDADC module:

- has an independent digital interface with register interfaces, control hardware, DMA and interrupt requests, and so on
- connects to a subset of external pins that also connect to an eQADC module
- connects to eTPU channels to support:
  - conversion result interrupt and DMA requests and flag to eTPU (with hardware gating)
  - high/low conversion limit checking with interrupt request and flag to eTPU

The following diagram shows the connections of an SDADC module to other on-chip components and external pins.



NOTE: SDADC\_4 has a 4:1 differential or 8:1 single-ended mux.

**Figure 27-2. SDADC connections to other modules**

# Chapter 28

## Sigma Delta Analog-to-Digital Converter (SDADC) Digital Interface

### 28.1 Chip-specific SDADC information

This chip has four instances of the SDADC module: SDADC\_1, SDADC\_2, and SDADC\_3, and SDADC\_4.

The following table summarizes varying features among the instances.

**Table 28-1. SDADC instance channels and muxes**

| Instance                     | # channels | Channels       | Single-ended input mux | Differential input mux |
|------------------------------|------------|----------------|------------------------|------------------------|
| SDADC1, SDADC_2, and SDADC_3 | 4          | AN[0] to AN[3] | 4:1                    | 2:1                    |
| SDADC_4                      | 8          | AN[0] to AN[7] | 8:1                    | 4:1                    |

#### 28.1.1 SDADC triggers and synchronous start

Each SDADC instance can be triggered to start a conversion either by a software write to [Software Trigger Key Register \(SDADC\\_STKR\)](#) or by a hardware trigger input selected in [Module Configuration Register \(SDADC\\_MCR\)](#) by writing the [TRIGSEL](#) field. See [Hardware triggering](#) for more information about triggering SDADC instances.

To support the synchronous start of all SDADC instances, the trigger output of each instance connects as a trigger input to every other instance.

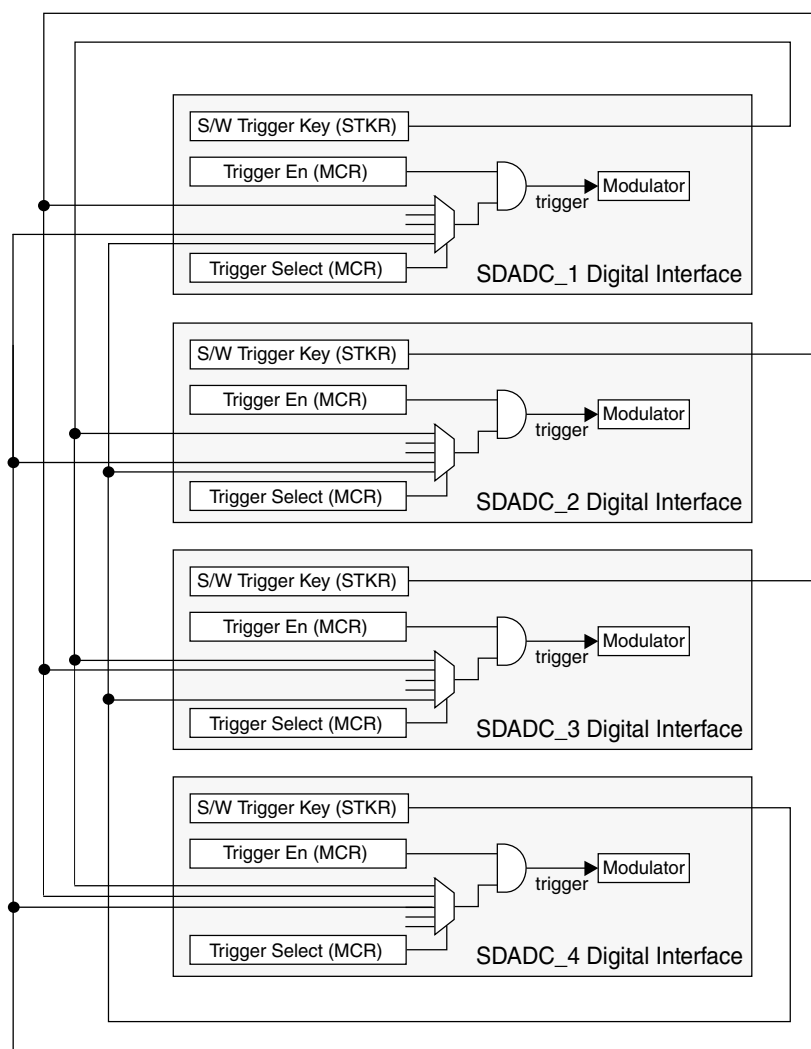


**Figure 28-1. SDADC simultaneous start connectivity**

Additional hardware trigger inputs connect to eTPU channels. The following table shows all of the hardware trigger inputs for each SDADC instance.

### Table 28-2. SDADC trigger inputs

| <b>MCR[TRIGSEL]<br/>value</b> | <b>SDADC_1 trigger input</b> | <b>SDADC_2 trigger input</b> | <b>SDADC_3 trigger input</b> | <b>SDADC_4 trigger input</b> |
|-------------------------------|------------------------------|------------------------------|------------------------------|------------------------------|
| 0000                          | Reserved                     | SDADC_1 trigger output       | SDADC_1 trigger output       | SDADC_1 trigger output       |
| 0001                          | SDADC_2 trigger output       | Reserved                     | SDADC_2 trigger output       | SDADC_2 trigger output       |
| 0010                          | SDADC_3 trigger output       | SDADC_3 trigger output       | Reserved                     | SDADC_3 trigger output       |
| 0011                          | SDADC_4 trigger output       | SDADC_4 trigger output       | SDADC_4 trigger output       | Reserved                     |
| 0100 - 0111                   | Reserved                     | Reserved                     | Reserved                     | Reserved                     |
| 1000                          | eTPU_A channel 20            | eTPU_A channel 22            | eTPU_A channel 24            | eTPU_A channel 26            |
| 1001                          | eTPU_A channel 21            | eTPU_A channel 23            | eTPU_A channel 25            | eTPU_A channel 27            |
| 1010                          | eTPU_B channel 20            | eTPU_B channel 22            | eTPU_B channel 24            | eTPU_B channel 26            |
| 1011                          | eTPU_B channel 21            | eTPU_B channel 23            | eTPU_B channel 25            | eTPU_B channel 27            |

*Table continues on the next page...*

**Table 28-2. SDADC trigger inputs (continued)**

| MCR[TRIGSEL] value | SDADC_1 trigger input | SDADC_2 trigger input | SDADC_3 trigger input | SDADC_4 trigger input |
|--------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 1100               | eTPU_C channel 15     | eTPU_C channel 17     | eTPU_C channel 19     | eTPU_C channel 21     |
| 1101               | eTPU_C channel 16     | eTPU_C channel 18     | eTPU_C channel 20     | eTPU_C channel 22     |
| 1110 - 1111        | Reserved              | Reserved              | Reserved              | Reserved              |

## 28.1.2 Connections among SDADCs and eTPUs

For information about SDADC and eTPU connections, see:

- [SDADC triggers and synchronous start](#)
- [SDADC connections to other modules](#)
- SIU details about the mapping of DMA/interrupt gating and mux advance inputs as well as the mapping of end-of-conversion and watchdog flag outputs

## 28.2 Introduction

The Sigma-Delta Analog-to-Digital Converter (SDADC) digital interface block controls the on-chip SDADC and holds control and status registers accessible for application. It provides accurate conversion data for a wide range of applications.

## 28.3 Overview

The SDADC digital interface block provides all control information for operating mode selection (single-ended or differential), analog input gain, decimation rate, high-pass filter enable, etc. It also generates the bias controls for common mode voltage selection, to bias the positive and negative terminals to different voltage levels (for example, half scale bias, reference ground, reference supply) for diagnostics. It generates the analog multiplexer control signals for the selected channels. To assist synchronous operation across multiple ADC blocks, hardware trigger support is implemented to start data conversion.

The following figure depicts the block diagram of an SDADC and its digital control block.

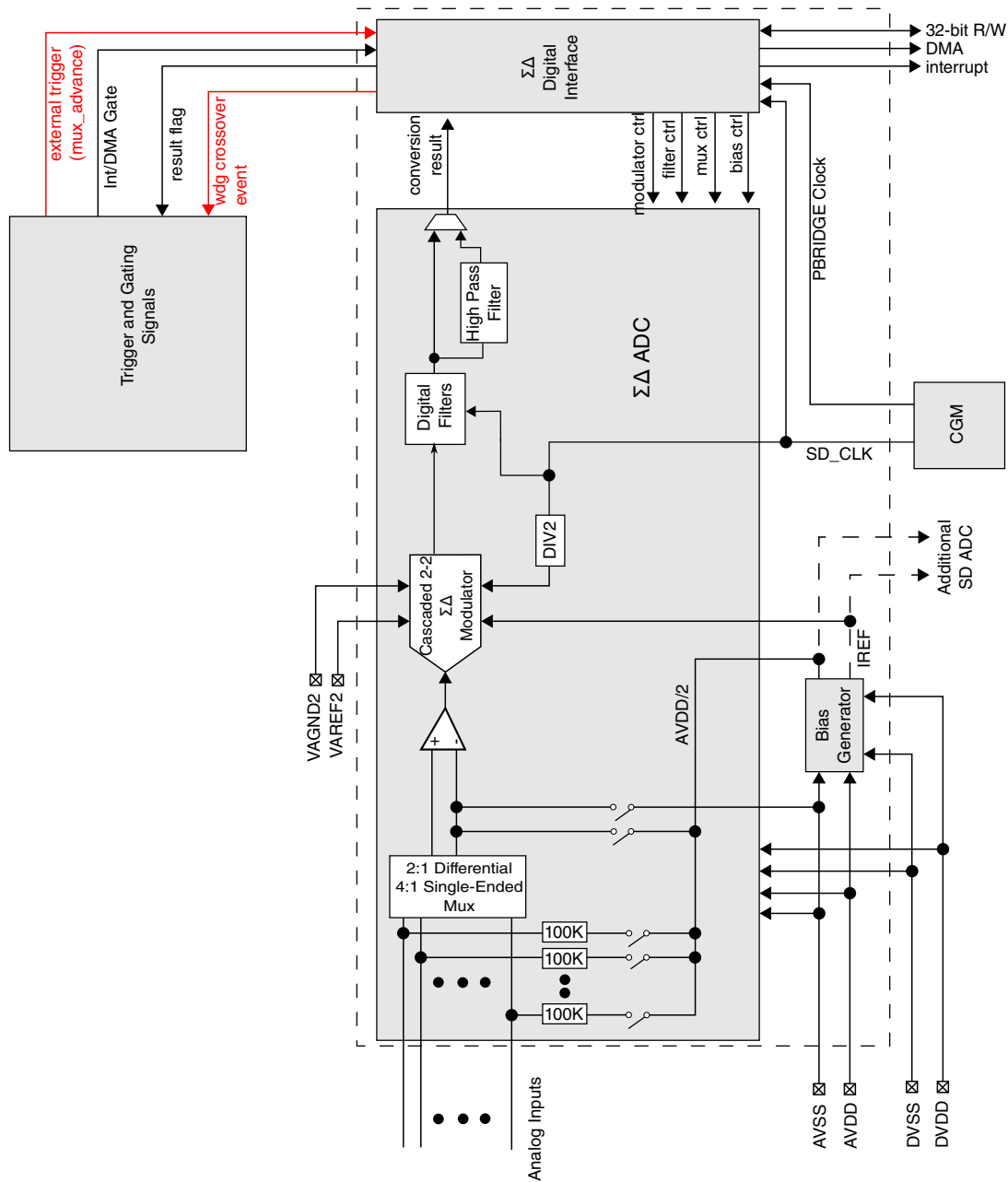


Figure 28-2. SDADC block diagram

## 28.4 Features

The features of the SDADC are as follows:

- 16-bit data resolution output
- Single-ended input or differential input mode of operation
- Programmable wraparound mechanism for both modes of operation



- Configurable trigger sources: hardware or software
- Configurable initial entry and wraparound values for the loop
- Configurable biasing for negative input terminal in single-ended mode
- Gain and offset calibration support using fixed bias for input terminals
- Programmable decimation rate
- Programmable gain for analog inputs
- Optional high-pass filter for pure AC application
- Hardware trigger support for synchronous operation of multiple SDADC blocks
- Trigger event output generation by software
- Programmable FIFO structure for storing up to 16 datawords of 16-bit converted data
- Interrupt/DMA request generation based on various conditions:
  - Data buffer above threshold
  - Data buffer overrun
  - Watchdog (WDG) crossover event
- Programmable threshold range watchdog support
- Low consumption mode
- Optional conversion process halt mechanism on SoC debug request

## 28.5 Modes of operation

This section describes the different operation modes of the SDADC. When exiting reset, the mode of operation is determined by the Module Configuration Register (MCR).

### 28.5.1 Differential input mode

This mode is entered by negating the MCR[MODE] bit. In this mode, a pair of analog inputs are connected to positive and negative terminals of ADC modulator. In order to support gain and offset calibration for diagnostics, it is possible in this mode to select fixed bias voltages to both input terminals.

### 28.5.2 Single-ended input mode

This mode is entered by asserting the MCR[MODE] bit. In this mode, the negative input terminal is biased with a fixed voltage based on selection information in MCR, and the other analog input is connected to the selected external analog channel.

## 28.6 External signal description

The following table shows the list of signals driven by external pins. See the chip-specific details about SDADC signals and package pinouts for information about signal muxing and availability external to the chip.

**Table 28-3. Signal properties**

| Name    | Function                                                          | I/O | Reset | Pull-up |
|---------|-------------------------------------------------------------------|-----|-------|---------|
| AN[0:7] | Single-ended analog inputs or Differential pairs of analog inputs | I   | —     | —       |
| VREFP   | Voltage reference for SDADC                                       | I   | —     | —       |
| VREFN   | Ground reference for SDADC                                        | I   | —     | —       |
| AVDD    | Analog voltage supply for SDADC                                   | I   | —     | —       |
| AVSS    | Analog ground supply for SDADC                                    | I   | —     | —       |
| DVDD    | Digital voltage supply for SDADC                                  | I   | —     | —       |
| DVSS    | Digital ground supply for SDADC                                   | I   | —     | —       |

## 28.7 Memory map and register definition

This section provides the memory map for the SDADC block.

**SDADC memory map**

| Address offset (hex) | Register name                                   | Width (in bits) | Access | Reset value | Section/ page               |
|----------------------|-------------------------------------------------|-----------------|--------|-------------|-----------------------------|
| 0                    | Module Configuration Register (SDADC_MCR)       | 32              | R/W    | 0000_0000h  | <a href="#">28.7.1/935</a>  |
| 4                    | Channel Selection Register (SDADC_CSR)          | 32              | R/W    | 0000_0000h  | <a href="#">28.7.2/938</a>  |
| 8                    | Reset Key Register (SDADC_RKR)                  | 32              | R/W    | 0000_A50Fh  | <a href="#">28.7.3/940</a>  |
| C                    | Status Flag Register (SDADC_SFR)                | 32              | R/W    | 0000_0100h  | <a href="#">28.7.4/941</a>  |
| 10                   | Request Select and Enable Register (SDADC_RSER) | 32              | R/W    | 0000_0000h  | <a href="#">28.7.5/943</a>  |
| 14                   | Output Settling Delay Register (SDADC_OSDR)     | 32              | R/W    | 0000_0000h  | <a href="#">28.7.6/945</a>  |
| 18                   | FIFO Control Register (SDADC_FCR)               | 32              | R/W    | 0000_0006h  | <a href="#">28.7.7/946</a>  |
| 1C                   | Software Trigger Key Register (SDADC_STKR)      | 32              | R/W    | 0000_0000h  | <a href="#">28.7.8/948</a>  |
| 20                   | Converted Data Register (SDADC_CDR)             | 32              | R      | 0000_0000h  | <a href="#">28.7.9/948</a>  |
| 24                   | WDG Threshold Register (SDADC_WTHHLR)           | 32              | R/W    | 0000_0000h  | <a href="#">28.7.10/949</a> |

## 28.7.1 Module Configuration Register (SDADC\_MCR)

The Module Configuration register (MCR) consists of different control bits to select the operating modes and various configuration settings which define the behavior of the SDADC block.

Address: 0h base + 0h offset = 0h

| Bit   | 0 | 1 | 2 | 3   | 4 | 5 | 6 | 7 | 8 | 9    | 10 | 11 | 12       | 13       | 14 | 15    |
|-------|---|---|---|-----|---|---|---|---|---|------|----|----|----------|----------|----|-------|
| R     | 0 |   |   | PDR |   |   |   |   | 0 | PGAN |    |    | Reserved | Reserved | 0  | HPFEN |
| W     |   |   |   |     |   |   |   |   |   |      |    |    |          |          |    |       |
| Reset | 0 | 0 | 0 | 0   | 0 | 0 | 0 | 0 | 0 | 0    | 0  | 0  | 0        | 0        | 0  | 0     |

| Bit   | 16    | 17        | 18 | 19     | 20      | 21 | 22 | 23 | 24  | 25 | 26 | 27      | 28     | 29    | 30   | 31 |
|-------|-------|-----------|----|--------|---------|----|----|----|-----|----|----|---------|--------|-------|------|----|
| R     | WDGEN | TRIGEDSEL |    | TRIGEN | TRIGSEL |    |    |    | FRZ | 0  |    | VCOMSEL | WRMODE | GECEN | MODE | EN |
| W     |       |           |    |        |         |    |    |    |     |    |    |         |        |       |      |    |
| Reset | 0     | 0         | 0  | 0      | 0       | 0  | 0  | 0  | 0   | 0  | 0  | 0       | 0      | 0     | 0    | 0  |

**SDADC\_MCR field descriptions**

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|----------|-------|----------|-------|----------|-------|----------|-------|----------|-------|----------|-------|----------|-------|----------|-------|----------|-------|----------|-------|----------|-------|----------|-------|----------|-------|----------|
| 0–2<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |
| 3–7<br>PDR      | Programmable Decimation Rate<br>This field selects the over-sampling ratio to be applied to support different passbands with a fixed input sampling clock. The output data rate $f_d$ is equal to $f_s / (2 \times \text{OSR})$ , where $f_s$ is the input sampling clock frequency.<br><br><table> <tr><td>00000</td><td>OSR = 24</td></tr> <tr><td>00001</td><td>OSR = 28</td></tr> <tr><td>00010</td><td>OSR = 32</td></tr> <tr><td>00011</td><td>OSR = 36</td></tr> <tr><td>00100</td><td>OSR = 40</td></tr> <tr><td>00101</td><td>OSR = 44</td></tr> <tr><td>00110</td><td>OSR = 48</td></tr> <tr><td>00111</td><td>OSR = 56</td></tr> <tr><td>01000</td><td>OSR = 64</td></tr> <tr><td>01001</td><td>OSR = 72</td></tr> <tr><td>01010</td><td>OSR = 75</td></tr> <tr><td>01011</td><td>OSR = 80</td></tr> <tr><td>01100</td><td>OSR = 88</td></tr> <tr><td>01101</td><td>OSR = 96</td></tr> </table> | 00000 | OSR = 24 | 00001 | OSR = 28 | 00010 | OSR = 32 | 00011 | OSR = 36 | 00100 | OSR = 40 | 00101 | OSR = 44 | 00110 | OSR = 48 | 00111 | OSR = 56 | 01000 | OSR = 64 | 01001 | OSR = 72 | 01010 | OSR = 75 | 01011 | OSR = 80 | 01100 | OSR = 88 | 01101 | OSR = 96 |
| 00000           | OSR = 24                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |
| 00001           | OSR = 28                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |
| 00010           | OSR = 32                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |
| 00011           | OSR = 36                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |
| 00100           | OSR = 40                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |
| 00101           | OSR = 44                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |
| 00110           | OSR = 48                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |
| 00111           | OSR = 56                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |
| 01000           | OSR = 64                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |
| 01001           | OSR = 72                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |
| 01010           | OSR = 75                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |
| 01011           | OSR = 80                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |
| 01100           | OSR = 88                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |
| 01101           | OSR = 96                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |       |          |

*Table continues on the next page...*

**SDADC\_MCR field descriptions (continued)**

| Field              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                    | 01110      OSR = 112<br>01111      OSR = 128<br>10000      OSR = 144<br>10001      OSR = 160<br>10010      OSR = 176<br>10011      OSR = 192<br>10100      OSR = 224<br>10101      OSR = 256<br>10110 - 11111    Reserved                                                                                                                                                                                                                                                                                        |
| 8<br>Reserved      | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 9–11<br>PGAN       | Programmable Gain<br>This field selects the gain to be applied to the analog input stage of the SDADC. The effective analog input becomes the input voltage level multiplied by the gain factor.<br><b>NOTE:</b> When SDADC_MCR[PGAN]=111b (programmable gain is 16), the least significant bit of the conversion data in SDADC_CDR[CDATA] is always 0.<br>000    Gain = 1<br>001    Gain = 2<br>010    Gain = 4<br>011    Gain = 8<br>100    Reserved<br>101    Reserved<br>110    Reserved<br>111    Gain = 16 |
| 12<br>Reserved     | This field is reserved.<br>This field is reserved and always has the value 0. Only the reset value shall be written to this register field.                                                                                                                                                                                                                                                                                                                                                                      |
| 13<br>Reserved     | This field is reserved.<br>This field is reserved and always has the value 0. Only the reset value shall be written to this register field.                                                                                                                                                                                                                                                                                                                                                                      |
| 14<br>Reserved     | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 15<br>HPFEN        | High Pass Filter Enable<br>0    High-pass (DC removal) filter is disabled<br>1    High-pass (DC removal) filter is enabled                                                                                                                                                                                                                                                                                                                                                                                       |
| 16<br>WDGEN        | Watchdog Enable<br>This bit enables the WDG monitor.<br>0    WDG is disabled<br>1    WDG is enabled                                                                                                                                                                                                                                                                                                                                                                                                              |
| 17–18<br>TRIGEDSEL | Trigger Edge Selection<br>00    Falling edge of trigger input is selected<br>01    Rising edge of trigger input is selected<br>10    Both edges of trigger input are selected<br>11    Both edges of trigger input are selected                                                                                                                                                                                                                                                                                  |

*Table continues on the next page...*

**SDADC\_MCR field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 19<br>TRIGEN      | <p>Trigger Enable</p> <p>This field enables the hardware trigger input to initiate a fresh conversion. Upon receiving a trigger event, SDADC reset input is asserted and deasserted synchronously with respect to the peripheral clock. The reset pulse width is fixed to four peripheral clock cycles. The trigger event is ignored if the SDADC internal modulator is not enabled (<math>MCR[EN] = 0</math>).</p> <p>0 Trigger input is disabled<br/>1 Trigger input is enabled</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 20–23<br>TRIGSEL  | <p>Trigger Input Selection</p> <p>This field selects which input will be used for hardware triggered conversions.</p> <p>0000 Input trigger 0 is selected. See chip-specific information for trigger source.<br/> 0001 Input trigger 1 is selected. See chip-specific information for trigger source.<br/> 0010 Input trigger 2 is selected. See chip-specific information for trigger source.<br/> 0011 Input trigger 3 is selected. See chip-specific information for trigger source.<br/> 0100 Input trigger 4 is selected. See chip-specific information for trigger source.<br/> 0101 Input trigger 5 is selected. See chip-specific information for trigger source.<br/> 0110 Input trigger 6 is selected. See chip-specific information for trigger source.<br/> 0111 Input trigger 7 is selected. See chip-specific information for trigger source.<br/> 1000 Input trigger 8 is selected. See chip-specific information for trigger source.<br/> 1001 Input trigger 9 is selected. See chip-specific information for trigger source.<br/> 1010 Input trigger 10 is selected. See chip-specific information for trigger source.<br/> 1011 Input trigger 11 is selected. See chip-specific information for trigger source.<br/> 1100 Input trigger 12 is selected. See chip-specific information for trigger source.<br/> 1101 Input trigger 13 is selected. See chip-specific information for trigger source.<br/> 1110 Input trigger 14 is selected. See chip-specific information for trigger source.<br/> 1111 Input trigger 15 is selected. See chip-specific information for trigger source.</p> |
| 24<br>FRZ         | <p>Freeze</p> <p>This field enables stopping the SDADC conversions at the end of the current channel conversion when the chip enters debug mode.</p> <p>When the chip enters debug mode, further conversions by the SDADC analog block will be stopped, and converted data output from the SDADC analog block are ignored. The ongoing channel conversion will be completed and the converted data is stored. When the chip exits debug mode, SDADC resumes the job that was left before halting the conversions.</p> <p>0 Conversions are not stopped<br/>1 Conversions are stopped</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 25–26<br>Reserved | <p>This field is reserved.<br/>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 27<br>VCOMSEL     | <p>Common Voltage Bias Selection</p> <p>This field selects the common voltage bias for the negative input terminal of the SDADC during single-ended mode (<math>MODE = 1</math>).</p> <p>0 Negative input terminal is biased with VREFN<br/>1 Negative input terminal is biased with VREFP/2 (half-scale bias)</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 28<br>WRMODE      | <p>Wrap-Around Mode</p> <p>This bit selects the wraparound mechanism for conversion of programmed sequence of channels.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

*Table continues on the next page...*

**SDADC\_MCR field descriptions (continued)**

| Field       | Description                                                                                                                                                                                                                                                                                                                                                   |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|             | 0 Wraparound mechanism disabled (in this case the default software control mechanism will be enabled)<br>1 Wraparound mechanism enabled                                                                                                                                                                                                                       |
| 29<br>GECEN | Accurate Gain Error Mode Enable<br><br>This bit selects the accurate gain error mode, and must be set during gain error calibration as well as during conversions requiring accurate gain error mode. It can be kept '0' if gain error calibration mode is not needed.<br><br>0 Gain error calibration mode disabled<br>1 Gain error calibration mode enabled |
| 30<br>MODE  | Mode selection<br><br>0 Differential input mode selected<br>1 Single-ended input mode selected                                                                                                                                                                                                                                                                |
| 31<br>EN    | Enable for SDADC block<br><br>0 SDADC internal modulator placed in low consumption mode<br>1 SDADC internal modulator enabled                                                                                                                                                                                                                                 |

**28.7.2 Channel Selection Register (SDADC\_CSR)**

The Channel Selection Register (CSR) selects analog inputs AN[0:7] to be connected to positive and negative terminals of the SDADC and controls the biasing of analog inputs to half-scale bias through 100 kOhm resistance.

Address: 0h base + 4h offset = 4h

|       |   |   |   |   |   |   |   |   |        |   |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|--------|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8      | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0 |   |   |   |   |   |   |   | BIASEN |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |        |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0      | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |    |    |    |    |              |    |    |    |    |    |    |         |    |    |    |
|-------|----|----|----|----|----|--------------|----|----|----|----|----|----|---------|----|----|----|
| Bit   | 16 | 17 | 18 | 19 | 20 | 21           | 22 | 23 | 24 | 25 | 26 | 27 | 28      | 29 | 30 | 31 |
| R     | 0  |    |    |    |    | ANCHSEL_WRAP |    |    | 0  |    |    |    | ANCHSEL |    |    |    |
| W     |    |    |    |    |    |              |    |    |    |    |    |    |         |    |    |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0            | 0  | 0  | 0  | 0  | 0  | 0  | 0       | 0  | 0  | 0  |

**SDADC\_CSR field descriptions**

| Field           | Description                                                                             |
|-----------------|-----------------------------------------------------------------------------------------|
| 0–7<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 8–15<br>BIASEN  | Bias enable                                                                             |

*Table continues on the next page...*

**SDADC\_CSR field descriptions (continued)**

| Field                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |         |                         |                         |                         |                         |   |   |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |   |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |   |     |     |       |       |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|-------------------------|-------------------------|-------------------------|-------------------------|---|---|-----|-------|-------|-----|-------|-------|-----|-------|-------|-----|-------|-------|-----|-------|-------|-----|-------|-------|-----|-------|-------|-----|-------|-------|---|-----|-------|---------|-----|-------|---------|-----|-------|---------|-----|-------|---------|-----|-------|---------|-----|-------|---------|-----|-------|---------|-----|-------|---------|---|-----|-----|-------|-------|
|                       | If BIASEN[i] = 1, analog input AN[i] (could be single-ended input or differential analog input) is connected to half-scale bias through a 100 kOhm resistor.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |         |                         |                         |                         |                         |   |   |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |   |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |   |     |     |       |       |
| 16–20<br>Reserved     | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |         |                         |                         |                         |                         |   |   |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |   |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |   |     |     |       |       |
| 21–23<br>ANCHSEL_WRAP | Analog Channel Selection Wraparound Value (ANCHSEL_WRAP)<br><br>When in wraparound mode, this field indicates the maximum value of the wraparound counter, after which it will wrap around back to initial value 000. This value should be programmed before enabling wraparound mechanism (MCR[WRMODE] = 1).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |         |                         |                         |                         |                         |   |   |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |   |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |   |     |     |       |       |
| 24–28<br>Reserved     | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |         |                         |                         |                         |                         |   |   |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |   |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |   |     |     |       |       |
| 29–31<br>ANCHSEL      | <p>Analog Channel Selection(ANCHSEL)</p> <p>Based on single-ended or differential mode of operation (MCR[MODE]), common mode voltage selection (MCR[VCOMSEL]), this field defines the connectivity of analog inputs to either positive or negative polarity terminals of the SDADC according to the following table.</p> <p>Wraparound Mode:</p> <ul style="list-style-type: none"><li>• This field also indicates the initial entry value for the first loop of the wraparound sequence from where the wraparound counter will start incrementing when wraparound mode is entered (when wraparound mechanism enabled (MCR[WRMODE] = 1)).</li><li>• If this field is written any time during wraparound mode (when MCR[WRMODE] = 1), the present value of the wraparound counter will be updated with the value of ANCHSEL - the next trigger will act on the updated channel, and the next increment will be on the updated channel. Here, the value of ANCHSEL chosen should be &lt; ANCHSEL_WRAP; otherwise it will be ignored by hardware.</li><li>• Wraparound will always be from initial counter value 000.</li></ul> <table><tr><th>MODE</th><th>VCOMSEL</th><th>ANCHSEL</th><th>INP (positive terminal)</th><th>INM (negative terminal)</th></tr><tr><td rowspan="16">1</td><td rowspan="8">0</td><td>000</td><td>AN[0]</td><td>VREFN</td></tr><tr><td>001</td><td>AN[1]</td><td>VREFN</td></tr><tr><td>010</td><td>AN[2]</td><td>VREFN</td></tr><tr><td>011</td><td>AN[3]</td><td>VREFN</td></tr><tr><td>100</td><td>AN[4]</td><td>VREFN</td></tr><tr><td>101</td><td>AN[5]</td><td>VREFN</td></tr><tr><td>110</td><td>AN[6]</td><td>VREFN</td></tr><tr><td>111</td><td>AN[7]</td><td>VREFN</td></tr><tr><td rowspan="8">1</td><td>000</td><td>AN[0]</td><td>VREFP/2</td></tr><tr><td>001</td><td>AN[1]</td><td>VREFP/2</td></tr><tr><td>010</td><td>AN[2]</td><td>VREFP/2</td></tr><tr><td>011</td><td>AN[3]</td><td>VREFP/2</td></tr><tr><td>100</td><td>AN[4]</td><td>VREFP/2</td></tr><tr><td>101</td><td>AN[5]</td><td>VREFP/2</td></tr><tr><td>110</td><td>AN[6]</td><td>VREFP/2</td></tr><tr><td>111</td><td>AN[7]</td><td>VREFP/2</td></tr><tr><td>0</td><td>0/1</td><td>000</td><td>AN[0]</td><td>AN[1]</td></tr></table> | MODE    | VCOMSEL                 | ANCHSEL                 | INP (positive terminal) | INM (negative terminal) | 1 | 0 | 000 | AN[0] | VREFN | 001 | AN[1] | VREFN | 010 | AN[2] | VREFN | 011 | AN[3] | VREFN | 100 | AN[4] | VREFN | 101 | AN[5] | VREFN | 110 | AN[6] | VREFN | 111 | AN[7] | VREFN | 1 | 000 | AN[0] | VREFP/2 | 001 | AN[1] | VREFP/2 | 010 | AN[2] | VREFP/2 | 011 | AN[3] | VREFP/2 | 100 | AN[4] | VREFP/2 | 101 | AN[5] | VREFP/2 | 110 | AN[6] | VREFP/2 | 111 | AN[7] | VREFP/2 | 0 | 0/1 | 000 | AN[0] | AN[1] |
| MODE                  | VCOMSEL                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | ANCHSEL | INP (positive terminal) | INM (negative terminal) |                         |                         |   |   |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |   |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |   |     |     |       |       |
| 1                     | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 000     | AN[0]                   | VREFN                   |                         |                         |   |   |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |   |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |   |     |     |       |       |
|                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 001     | AN[1]                   | VREFN                   |                         |                         |   |   |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |   |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |   |     |     |       |       |
|                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 010     | AN[2]                   | VREFN                   |                         |                         |   |   |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |   |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |   |     |     |       |       |
|                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 011     | AN[3]                   | VREFN                   |                         |                         |   |   |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |   |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |   |     |     |       |       |
|                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 100     | AN[4]                   | VREFN                   |                         |                         |   |   |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |   |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |   |     |     |       |       |
|                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 101     | AN[5]                   | VREFN                   |                         |                         |   |   |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |   |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |   |     |     |       |       |
|                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 110     | AN[6]                   | VREFN                   |                         |                         |   |   |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |   |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |   |     |     |       |       |
|                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 111     | AN[7]                   | VREFN                   |                         |                         |   |   |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |   |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |   |     |     |       |       |
|                       | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 000     | AN[0]                   | VREFP/2                 |                         |                         |   |   |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |   |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |   |     |     |       |       |
|                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 001     | AN[1]                   | VREFP/2                 |                         |                         |   |   |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |   |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |   |     |     |       |       |
|                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 010     | AN[2]                   | VREFP/2                 |                         |                         |   |   |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |   |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |   |     |     |       |       |
|                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 011     | AN[3]                   | VREFP/2                 |                         |                         |   |   |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |   |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |   |     |     |       |       |
|                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 100     | AN[4]                   | VREFP/2                 |                         |                         |   |   |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |   |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |   |     |     |       |       |
|                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 101     | AN[5]                   | VREFP/2                 |                         |                         |   |   |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |   |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |   |     |     |       |       |
|                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 110     | AN[6]                   | VREFP/2                 |                         |                         |   |   |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |   |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |   |     |     |       |       |
|                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 111     | AN[7]                   | VREFP/2                 |                         |                         |   |   |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |   |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |   |     |     |       |       |
| 0                     | 0/1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 000     | AN[0]                   | AN[1]                   |                         |                         |   |   |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |     |       |       |   |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |     |       |         |   |     |     |       |       |

Table continues on the next page...

**SDADC\_CSR field descriptions (continued)**

| Field | Description |         |         |                         |                         |
|-------|-------------|---------|---------|-------------------------|-------------------------|
|       | MODE        | VCOMSEL | ANCHSEL | INP (positive terminal) | INM (negative terminal) |
|       |             |         | 001     | AN[2]                   | AN[3]                   |
|       |             |         | 010     | AN[4]                   | AN[5]                   |
|       |             |         | 011     | AN[6]                   | AN[7]                   |
|       |             |         | 100     | VREFN                   | VREFN                   |
|       |             |         | 101     | VREFP/2                 | VREFP/2                 |
|       |             |         | 110     | VREFP                   | VREFN                   |
|       |             |         | 111     | VREFN                   | VREFP                   |

**28.7.3 Reset Key Register (SDADC\_RKR)**

The Reset Key Register (RKR) is the target for a predefined key write operation used to reset the SDADC to start a fresh conversion.

Address: 0h base + 8h offset = 8h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|-----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15        | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    | RESET_KEY |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0         | 1  | 0  | 1  | 0  | 0  | 1  | 0  | 1  | 0  | 0  | 0  | 0  | 1  | 1  | 1  | 1  |

**SDADC\_RKR field descriptions**

| Field              | Description                                                                                                                                                                                                                                                                                                               |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                   |
| 16–31<br>RESET_KEY | Reset Key<br>This field, when written with 5AF0h, is used to generate the reset for the SDADC block and to reload the internal counter with the start value programmed in OSDR. Any write access to this register which is different from the predefined key is ignored. Read access will always return the inverted key. |



## 28.7.4 Status Flag Register (SDADC\_SFR)

The Status Flag Register (SFR) contains status and flag bits. These bits are set by hardware and reflect the status of the SDADC and indicate the occurrence of events that may generate an interrupt or a DMA request. Software can clear some flags by writing '1' to those bits. Writing a '0' to a flag has no effect.

Address: 0h base + Ch offset = Ch

|       |   |   |   |   |   |   |   |   |   |   |    |    |             |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|-------------|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12          | 13 | 14 | 15 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    | ANCHSEL_CNT |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |             |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0           | 0  | 0  | 0  |

|       |    |    |    |    |    |    |    |      |    |    |    |    |      |      |      |       |      |
|-------|----|----|----|----|----|----|----|------|----|----|----|----|------|------|------|-------|------|
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23   | 24 | 25 | 26 | 27 | 28   | 29   | 30   | 31    |      |
| R     | 0  |    |    |    |    |    |    | DFEF | 0  |    |    |    | WTHH | WTHL | CDVF | DFORF | DFFF |
| W     |    |    |    |    |    |    |    |      |    |    |    |    | w1c  | w1c  |      | w1c   | w1c  |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1    | 0  | 0  | 0  | 0  | 0    | 0    | 0    | 0     |      |

**SDADC\_SFR field descriptions**

| Field                | Description                                                                                                                                                                                                                      |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–12<br>Reserved     | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                          |
| 13–15<br>ANCHSEL_CNT | Analog Channel Selection Counter(ANCHSEL_CNT)<br>This field reflects the current value of the internal Analog Channel Selection Counter. It indicates the present number of the channel selected through the analog multiplexer. |
| 16–22<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                          |
| 23<br>DFEF           | Data FIFO Empty Flag<br>This field is asserted when there is no dataword in the FIFO.                                                                                                                                            |

*Table continues on the next page...*

## SDADC\_SFR field descriptions (continued)

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | 0 Data FIFO is not empty<br>1 Data FIFO is empty                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 24–26<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 27<br>WTHH        | Watchdog Upper Threshold Cross Over Event<br>This field is set when Converted Data crosses Upper Threshold of WDG. It can be cleared by software writing '1' to clear.<br><b>NOTE:</b> This bit is also cleared by an acknowledgement from the DMA module when all of the following conditions are met: <ul style="list-style-type: none"> <li>• SDADC_RSER[WTHDIRE]=1 (When this field is set by the occurrence of a watchdog upper threshold cross over event, a request is generated)</li> <li>• SDADC_SFR[WTHDIRS]=1 (The request generated when this field is set is a DMA request)</li> <li>• SDADC_RSER[GDIGE]=0 OR, if SDADC_RSER[GDIGE]=1, the external gating signal is asserted</li> </ul> 0 Watchdog Upper Threshold Cross Over Event did not occur<br>1 Watchdog Upper Threshold Cross Over Event occurred |
| 28<br>WTHL        | Watchdog Lower Threshold Cross Over Event<br>This field is set when Converted Data crosses Lower Threshold of WDG. It can be cleared by software writing '1' to clear.<br><b>NOTE:</b> This bit is also cleared by an acknowledgement from the DMA module when all of the following conditions are met: <ul style="list-style-type: none"> <li>• SDADC_RSER[WTHDIRE]=1 (When this field is set by the occurrence of a watchdog lower threshold cross over event, a request is generated)</li> <li>• SDADC_SFR[WTHDIRS]=1 (The request generated when this field is set is a DMA request)</li> <li>• SDADC_RSER[GDIGE]=0 OR, if SDADC_RSER[GDIGE]=1, the external gating signal is asserted</li> </ul> 0 Watchdog Lower Threshold Cross Over Event did not occur<br>1 Watchdog Lower Threshold Cross Over Event occurred |
| 29<br>CDVF        | Converted Data Valid Flag<br>This field gives the actual status of data coming from the SDADC block. The converted data is not valid or is corrupted whenever there is a change in configuration settings determined by MCR, channel selection programmed in CSR. Any write access to the registers MCR and CSR will reset this flag. This flag is set automatically after an internal timer counts a number of output clock fd clock cycles specified by the value of SDADC_OSDR[OSD].<br>0 Data output from SDADC is not valid<br>1 Data output from SDADC is valid                                                                                                                                                                                                                                                   |
| 30<br>DFORF       | Data FIFO Overrun Flag<br>This field indicates that software or DMA failed to prevent FIFO from overflowing with converted datawords. Software can clear this bit by writing '1' to it after reading the FIFO contents through CDR. This flag remains set if software or DMA failed to clear the condition that caused this flag setting. If DFORF is set, further datawords will not be received into FIFO even if sufficient room exists.<br>0 No overrun has occurred since the last time the flag was cleared<br>1 Overrun has occurred or the DFORF has not been cleared since the last overrun occurred                                                                                                                                                                                                           |

Table continues on the next page...

**SDADC\_SFR field descriptions (continued)**

| Field      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31<br>DFFF | <p>Data FIFO Full Flag</p> <p>This bit is set when the number of converted datawords in the FIFO is equal to or more than the number indicated by FCR[FTHL D]. Software can clear this bit by writing '1' to it. This bit is also cleared by an acknowledgement from the DMA controller when DMA request generation is selected. This bit remains set if software or DMA failed to clear the condition that caused this flag setting. Even if DFFF is set, datawords will continue to be received until an overrun condition occurs. Datawords will not be received if the converted data is not valid (CDVF = 0).</p> <p><b>NOTE:</b> Whenever MCR[EN] is cleared in order to stop the SDADC or change the channel configuration, it is necessary to also clear the RSER[DFFDIRE] bit alongwith. Similarly if DMA or software is too slow in not reading the FIFO datawords resulting in DFFF condition it is advisable to stop the SDADC by clearing the MCR[EN] followed by clearing of RSER[DFFDIRE] bit. Both these action will ensure safe operation.</p> <p>0 The number of datawords in FIFO is less than the number indicated by FCR[FTHL D]<br/> 1 The number of datawords in FIFO is equal to or greater than the number indicated by FCR[FTHL D] at some point in time since this flag was last cleared</p> |

**28.7.5 Request Select and Enable Register (SDADC\_RSER)**

The Request Select and Enable Register (RSER) serves two purposes:

- Enables the flag bits in SFR to generate DMA requests or interrupt requests.
- Selects the type of request to generate.

Refer to the field descriptions for the type of requests that are supported.

Address: 0h base + 10h offset = 10h

|       |       |    |    |    |    |    |    |    |    |    |    |    |         |         |         |         |
|-------|-------|----|----|----|----|----|----|----|----|----|----|----|---------|---------|---------|---------|
| Bit   | 0     | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12      | 13      | 14      | 15      |
| R     | 0     |    |    |    |    |    |    |    |    |    |    |    |         | WTHDIRS | DFFDIRS |         |
| W     |       |    |    |    |    |    |    |    |    |    |    |    |         |         |         |         |
| Reset | 0     | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0       | 0       | 0       | 0       |
| Bit   | 16    | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28      | 29      | 30      | 31      |
| R     | GDIGE | 0  |    |    |    |    |    |    |    |    |    |    | WTHDIRE | CDVEE   | DFORIE  | DFFDIRE |
| W     |       |    |    |    |    |    |    |    |    |    |    |    |         |         |         |         |
| Reset | 0     | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0       | 0       | 0       | 0       |

**SDADC\_RSER field descriptions**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–13<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                        |
| 14<br>WTHDIRS     | WDG Threshold Cross Over Event DMA/Interrupt Request Select<br>When SFR[WTHH/WTHL] and RSER[WTHDIRE] fields are set, this field selects between generating a DMA request or an interrupt request.<br><br>0 Interrupt request is selected<br>1 DMA request is selected                                                                                                                                                                                                          |
| 15<br>DFFDIRS     | Data FIFO Full DMA/Interrupt Request Select<br>When the SFR[DFFF] and SFR[DFFDIRE] fields are set, this field selects between generating a DMA request or an interrupt request.<br><br>0 Interrupt request is selected<br>1 DMA request is selected                                                                                                                                                                                                                            |
| 16<br>GDIGE       | Global DMA/Interrupt Gating Enable<br>This field determines whether all module interrupts and DMA requests need to be gated by an external gating signal or not.<br><br>0 No impact of external gating signal on module DMA/interrupt requests<br>1 All module DMA/interrupt requests are qualified only when external gating signal is asserted                                                                                                                               |
| 17–27<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                        |
| 28<br>WTHDIRE     | WDG Threshold Cross Over Event DMA/Interrupt Request Enable<br>This field enables the SFR[WTHH/WTHL] field to generate a request. The WTHDIRS field determines which is selected, a DMA request or an interrupt request. The final DMA/interrupt request also depends on the gating signal if enabled by the GDIGE field.<br><br>0 Interrupt/DMA request is disabled on WDG Threshold Cross Over Event<br>1 Interrupt/DMA request is enabled on WDG Threshold Cross Over Event |
| 29<br>CDVEE       | Conversion Data Valid Event Enable<br>The status of SFR[CDVF] gated with RSER[CDVEE] is passed on as an event on “conversion_flag” output port, which can be used at SoC level for indication of valid data conversion window. Once enabled, the event output will remain asserted as long as valid data conversions are ongoing.<br><br>0 Event output disabled<br>1 Event output assertion/deassertion based on the SFR[CDVF] field                                          |
| 30<br>DFORIE      | Data FIFO Overrun Interrupt Enable<br>This bit enables the SFR[DFORF] field to generate an interrupt request. The final interrupt request also depends on the gating signal if enabled by the GDIGE field.<br><br>0 Interrupt request is disabled when data FIFO overrun condition occurs<br>1 Interrupt request is enabled when data FIFO overrun condition occurs                                                                                                            |
| 31<br>DFFDIRE     | Data FIFO Full DMA/Interrupt Request Enable<br>This field enables the SFR[DFFF] field to generate a request. The DFFDIRS field determines which is selected, a DMA request or an interrupt request. The final DMA/interrupt request also depends on the gating signal if enabled by the GDIGE field.                                                                                                                                                                           |

*Table continues on the next page...*

**SDADC\_RSER field descriptions (continued)**

| Field | Description                                                            |
|-------|------------------------------------------------------------------------|
| 0     | Interrupt/DMA request is disabled when data FIFO full condition occurs |
| 1     | Interrupt/DMA request is enabled when data FIFO full condition occurs  |

**28.7.6 Output Settling Delay Register (SDADC\_OSDR)**

The Output Settling Delay Register (OSDR) provides a delay value to qualify the converted output data coming from the SDADC.

Address: 0h base + 14h offset = 14h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23  | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    | OSD |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**SDADC\_OSDR field descriptions**

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–23<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 24–31<br>OSD     | Output Settling Delay<br>This field defines the delay to qualify the conversion data stored in CDR. Whenever the SDADC block is reset in order to start the conversion from a fresh state, an internal timer is loaded with this start value. The counter counts down with output clock $f_d$ until it reaches zero, and then it generates a flag which qualifies the converted data. The output clock $f_d$ is equal to $f_s / (2 \times \text{OSR})$ , where $f_s$ is the input sampling clock frequency. See the electrical specifications for the settling delays.<br><br><b>NOTE:</b> Refer to the $t_{\text{settling}}$ and $t_{\text{LATENCY}}$ parameters in the device datasheet for the minimum allowed output settling delay values. |

28.7.7 FIFO Control Register (SDADC\_FCR)

The FIFO Control Register (FCR) provides the ability to enable or disable the FIFO functionality, and to control the amount of available FIFO depth to be used.

NOTE

1. It is important to note that the FIFO depth is set at the factory and cannot be changed but the amount of available FIFO space actually used can be set by specifying the value in the FTHLD field. The FSIZE field indicates the amount of FIFO space available. By setting the value of FTHLD to be less than the FIFO depth indicated by FSIZE, the FIFO is considered to be full when the number of data words in the FIFO is equal or greater than FTHLD.
2. This register may be read at any time, but should only be written when SDADC is disabled (MCR[EN] = 0).

Address: 0h base + 18h offset = 18h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |      |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|------|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15   |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    | 0    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    | FRST |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0    |

|       |    |    |    |    |       |    |    |    |    |    |    |    |       |    |       |    |
|-------|----|----|----|----|-------|----|----|----|----|----|----|----|-------|----|-------|----|
| Bit   | 16 | 17 | 18 | 19 | 20    | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28    | 29 | 30    | 31 |
| R     | 0  |    |    |    |       |    |    |    |    |    |    |    |       |    | FSIZE |    |
| W     |    |    |    |    | FTHLD |    |    |    |    |    |    |    | FOWEN |    |       | FE |
| Reset | 0  | 0  | 0  | 0  | 0     | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0     | 1  | 1     | 0  |

**SDADC\_FCR field descriptions**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                        |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–14<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                            |
| 15<br>FRST        | FIFO Flush Reset<br><br>This is functional soft reset to flush the current FIFO content and initialize the internal FIFO pointers. This is W1S(write one shot) bit which can be written as '1' but always read as '0'. Use this option only after disabling the ADC i.e. MCR[EN]='0'.<br><br>0 No effect.<br>1 Generate a single cycle reset event to flush the FIFO.                                              |
| 16–19<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                            |
| 20–23<br>FTHLD    | FIFO Threshold<br><br>When the number of datawords in the data FIFO is greater than the value in FTHLD field, the FIFO full event is flagged. An interrupt or a DMA request will be generated as determined by DFFDIRE and DFFDIRS fields of RSER. For proper operation, the value in the FTHLD field must be set to be less than or equal to the FIFO size as indicated by FSIZE.                                 |
| 24–27<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                            |
| 28<br>FOWEN       | FIFO Over Write Enable<br><br>When set the built-in FIFO structure allows the overwriting of new converted data over older converted data after the FIFO has number of entries equal to the FSIZE. In this case overrun will not be seen.<br><br>0 Data FIFO OW option disabled.<br>1 Data FIFO OW option enabled.                                                                                                 |
| 29–30<br>FSIZE    | FIFO Size<br><br>The maximum number of converted datawords that can be stored in the data FIFO before an overrun occurs. This field is read-only. The reset value of this field depends on the implementation parameter to define the FIFO size.<br><br>00 FIFO depth = 1 dataword<br>01 FIFO depth = 4 datawords<br>10 FIFO depth = 8 datawords<br>11 FIFO depth = 16 datawords                                   |
| 31<br>FE          | FIFO Enable<br><br>When this field is set the built-in FIFO structure is enabled. The size of the FIFO structure is indicated by the FSIZE field. If this field is not set then the FIFO depth is set to one dataword regardless of the value in the FSIZE field.<br><br>0 Data FIFO is not enabled for multiple datawords; FIFO depth is one dataword<br>1 Data FIFO is enabled; FIFO depth is indicated by FSIZE |

## 28.7.8 Software Trigger Key Register (SDADC\_STKR)

The Software Trigger Key Register (STKR) allows generating the trigger event output by a software write operation.

Address: 0h base + 1Ch offset = 1Ch

| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16     | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|--------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | ST_KEY |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0      | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

### SDADC\_STKR field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                       |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                           |
| 16–31<br>ST_KEY  | Software Trigger Key<br>This bitfield, when written with FFFFh, is used to generate a trigger event output which can be used to trigger conversions of multiple SDADC blocks synchronously depending on the chip's implementation. Any write access to this register which is different from the predefined key is ignored. Read access will always return 0000h. |

## 28.7.9 Converted Data Register (SDADC\_CDR)

The Converted Data Register (CDR) stores the converted data in the data FIFO.

The conversion result in the CDATA word is a 16-bit, signed value.

### NOTE

When SDADC\_MCR[PGAN]=111b (programmable gain is 16), the least significant bit of the conversion data in SDADC\_CDR[CDATA] is always 0.

Address: 0h base + 20h offset = 20h

| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16    | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | CDATA |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0     | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |



**SDADC\_CDR field descriptions**

| Field            | Description                                                                                                                                                                                                  |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                      |
| 16–31<br>CDATA   | Converted Data Register (CDATA[13:0])<br>The converted datawords can be read from FIFO by reading this register. The data width is 16 bits which has actual 16-bit data (bits 15 down to 0) coming from ADC. |

**28.7.10 WDG Threshold Register (SDADC\_WTHHLR)**

This is the Converted Data Range WDG Threshold register.

**NOTE**

A single WDG monitor will check for lower and upper threshold crossover events.

Address: 0h base + 24h offset = 24h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**SDADC\_WTHHLR field descriptions**

| Field         | Description                                                                                                                                                                        |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>THRH  | WDG Upper Threshold Value<br>This indicates the upper threshold for the converted data on any channel, above which threshold a crossover event would be flagged through SFR[WTHH]. |
| 16–31<br>THRL | WDG Lower Threshold Value<br>This indicates the lower threshold for the converted data on any channel, below which threshold cross over event would be flagged through SFR[WTHL].  |

**28.8 Functional description**

The SDADC has the following distinct available modes determined by fields in MCR:

- Differential input mode
- Single-ended input mode
- Low consumption mode

### 28.8.1 Differential input mode

Differential input mode is entered when the SDADC\_MCR[MODE] field is set to '0' and the SDADC\_MCR[EN] field is set to '1'. In differential input mode, a pair of analog inputs among AN[0:7] is connected to the input terminals of the SDADC modulator block. The module supports four differential pairs: AN[0,1], AN[2,3], AN[4,5], and AN[6,7].

The modulator block has two input terminals, positive (INP) and negative (INM). The selection of input pair depends on the SDADC\_CSR[ANCHSEL] field. The connectivity of each analog input in a given differential pair, whether it is connected to positive or negative terminal, is determined according to the Analog input AN[0:7] selection table in the Channel Selection Register description.

The differential input mode can also be used to perform gain and offset calibration, where the input terminals are biased with fixed voltages coming from the bias block.

### 28.8.2 Single-ended input mode

Single-ended input mode is entered by setting the SDADC\_MCR[MODE] and SDADC\_MCR[EN] fields to '1'. In this mode, one of the analog inputs among AN[0:7] is selected via the SDADC\_CSR[ANCHSEL] field and is connected to the positive terminal. The negative input terminal is biased with a fixed internal voltage. There are two internal bias voltages available: reference ground (VREFN) and half-scale bias (VREFP/2). The selection of this common mode voltage is controlled via the SDADC\_MCR[VCOMSEL] field.

### 28.8.3 Low consumption mode

This is the default mode after exiting from reset. This mode is exited by asserting the MCR[EN] field to enable the SDADC block. When data conversion is not required, the SDADC internal modulator can be put in low consumption mode to reduce power consumption.

### 28.8.4 Analog input multiplexing and bias support

The complete selection and connectivity of analog inputs AN[0:7] is based on the ANCHSEL field of CSR and the MODE and VCOMSEL bits of MCR, according to the Analog input AN[0:7] selection table in the Channel Selection Register description. Each analog input can be biased to half scale ( $AVDD/2$ ) via on-chip 100 k $\Omega$  resistance. This allows simple AC coupling to the external analog inputs through a capacitor. This biasing is applicable irrespective of whether analog channel is selected through multiplexer or not.

### 28.8.5 Programmable gain and decimation rate

All analog inputs can be configured to have a selectable input gain as defined in the PGAN field description in MCR. This means the input signal is sampled and the result is amplified by the factor determined by PGAN before providing the same to modulator.

The SDADC module is provided with a fixed input sampling clock  $f_s$ . The input signal is grossly oversampled by the modulator in order to reduce quantization noise. To support different passbands with a fixed sampling clock, a programmable decimation rate is implemented. The process of decimation, to eliminate redundant data at the output while retaining the necessary information, is controlled by the field PDR in MCR.

### 28.8.6 High-pass filter support

For pure AC applications, it is useful to remove any DC component in the input signal. An optional high-pass filter is implemented which can be enabled by asserting the MCR[HPFEN] field. The  $-3$  dB frequency of the filter is fixed (not programmable) and is specified in the device data sheet. This high-pass filter is implemented in the decimation filter logic of the SDADC.

### 28.8.7 Data conversion

The SDADC operates in a continuous conversion mode as soon as it is enabled. There is an inherent settling time required after SDADC startup before conversion data is considered valid.

#### NOTE

During that period, the Converted Data Valid Flag bit (SDADC\_SFR[CDVF]) reads '0'.

Settling time duration depends on the latency and output settling time specifications. To account for the settling time, an internal timer is implemented that counts down from a start value determined by the value of the SDADC\_OSDR[OSD] field.

The internal countdown timer is triggered by an SDADC reset, i.e., by writing the value 5AF0h to the SDADC\_RKR[RESET\_KEY] field.

### NOTE

An SDADC reset is required any time there is a change to the SDADC configuration (via changes to the SDADC\_MCR register settings), or a change to analog channel selection (via SDADC\_CSR register settings).

When the timer reaches '0', the SDADC\_SFR[CDVF] field is asserted and converted data is loaded into data FIFO on every rising edge of the output clock ( $f_d$ ).

Any write access to the SDADC\_MCR or SDADC\_CSR register causes the SDADC\_SFR[CDVF] flag to be cleared, indicating that data coming from SDADC is not valid. If the SDADC\_SFR[CDVF] flag is '0', the following flags are blocked from becoming asserted:

- Data FIFO Full Flag (SDADC\_SFR[DFFF])
- Data FIFO Overrun Flag (SDADC\_SFR[DFORF])

If the SDADC\_SFR[DFFF] bit is already set, the data FIFO is already full before SDADC\_SFR[CDVF] deassertion.

Wraparound Control Mechanism: Here a wraparound logic includes all analog channels of the input multiplexer from  $Ain("000")$  to  $Ain(CSR[ANCHSEL\_WRAP])$  in both Single ended and Differential ended mode in wraparound sequence triggered by hardware or software source:

- MCR[WRMODE] bit would enable or disable the wraparound mechanism. Clearing the WRMODE bit automatically switches FSM back to Software control mode.
- Every valid trigger advances the wraparound counter to next channel to be converted.
- Wraparound Counter: An internal wraparound counter which points to analog channel to be converted takes the initial entry value for the first loop of execution from CSR[ANCHSEL] while entering wraparound mode. From next loop onwards the counter wraparounds to default initial value "000". Maximum value of the counter (wraparound value) should be programmed in CSR[ANCHSEL\_WRAP] before entering wraparound mode. The definition of ANCHSEL and ANCHSEL\_WRAP is provided in the Analog input AN[0:7] selection table in the Channel Selection Register description.
- Upon entering or during wraparound mode user should take care to program ANCHSEL value less than ANCHSEL\_WRAP value; otherwise hardware would reject the new ANCHSEL value.

- Possible trigger sources:
  - SOURCE1 (Software Bit): writing to STKR.ST\_KEY at any time during the wraparound mode (irrespective of MCR[TRIGEN or TRIGSEL] setting) generates a trigger pulse. Each such trigger initiates a fresh conversion and advances the wraparound counter by one.

### NOTE

Care should be taken while programming the TRIGEN/TRIGSEL of each SDADC instance that no unintended simultaneous SW triggering of multiple instances happen.

- SOURCE2 (Hardware Trigger): Here external trigger output enabled by MCR[TRIGEN or TRIGEDSEL or TRIGSEL] before entering wraparound mode for selective or all SDADC instances provides trigger. Each such valid trigger initiates a fresh conversion and advances the wraparound counter by one. One of the hw\_trig\_in ports can be used to connect the external trigger output.
- Next channel number to be converted can be updated by software even in the middle of wraparound sequence execution. This can be done by rewriting CSR[ANCHSEL]. After CSR[ANCHSEL] is rewritten, the next trigger takes updated value and the wraparound counter would increment from the updated value.
- Current wraparound counter value can be read from the SFR[ANCHSEL\_CNT] bitfield.

## 28.8.8 Hardware triggering

Software Control Mechanism: SDADC conversion can be triggered by various hardware events coming from either external pins or internal timers. The field TRIGSEL of MCR selects which hardware event is to be used for triggering conversions. This trigger event, which is synchronous to the peripheral clock, is enabled by asserting MCR[TRIGEN]. As soon as a trigger event is detected, reset input of SDADC block is asserted synchronously for four cycles. This will ensure that SDADC starts from a fresh state when reacting to the input trigger event. The SDADC internal modulator must be enabled before enabling the trigger event — otherwise the trigger event will be ignored.

If the same trigger event is provided to multiple SDADC digital interface modules and the respective enable controls (TRIGEN bit MCR) are asserted, synchronous operation of multiple SDADC blocks can be achieved with output results being updated at the same time. Once the reset input is deasserted, the first valid conversion output is stored in data FIFO after a configurable delay determined by OSD[OSD].

It is also possible to generate a software trigger event output by a write access of STKR with a predefined key. This event output can be used as a trigger to achieve synchronous software start of multiple SDADC blocks.

In case of a hardware trigger, the skew/jitter will be less than 100 ns. It will be more than one ADC clock cycle due to synchronization. In case of a software trigger, all SDADCs start synchronous sampling.

Wraparound Control Mechanism: In wraparound control mode, hardware and software triggers are equally valid. The hardware trigger source should be selected before enabling the wraparound mode.

### **28.8.9 Interrupt/DMA request support**

The SDADC has one condition that generates interrupts only and another that generates interrupts or DMA request.

The DFFF field is asserted when the data FIFO full condition is detected, indicating that the number of converted datawords stored in data FIFO is equal to or greater than the value indicated by threshold determined by FCR[FTHLD]. The interrupt or DMA request is generated only when RSER[DFFDIRE] is asserted. RSER[DFFDIRS] selects whether a DMA request or an interrupt request is generated.

The data FIFO overrun flag (DFORF) indicates the FIFO overflow condition. Further converted datawords cannot be received and data is lost. The interrupt request is generated only when RSER[DFORIE] is asserted.

Apart from above, a WDG threshold crossover event can also generate an interrupt or DMA request.

If RSER[GDIGE] is set, all module interrupt/DMA requests are qualified only when an external global gating signal is asserted.

### **28.8.10 Gain calibration support**

To perform gain calibration, the following sequence needs to be applied.

1. Select differential mode of operation by writing MCR[MODE] to '0'.
2. Enable gain error calibration mode by writing MCR[GECEN] to '1'.
3. Configure the mux selection ANCHSEL field of CSR to '110'. This configuration will apply VREFP on positive terminal and VREFN on negative terminal.
4. Disable the bias on all input analog channels by writing ENBIAS field of CSR to 00h.

5. Disable the high-pass filter by deasserting MCR[HPFEN].
6. Generate a reset event by writing 5AF0h to RESET\_KEY of RKR.
7. Read the digital output (CDR[CDATA]) stored in FIFO after the output settling time. Expected output if there is no gain error is 0111\_1001\_1001\_1001b, corresponding to full positive scale after attenuation inserted by the internal filter ( $1 \times 0.95$ ). The measurement should be repeated to reduce contribution of noise during the calibration process.  $D_p$  is the average value of attenuated positive full scale given by  $D_p = \text{AVERAGE}(\text{CDR}[\text{CDATA}])$ .
8. Change the mux selection ANCHSEL field of CSR to '111'. This configuration will apply VREFN on positive terminal and VREFP on negative terminal.
9. Generate a reset event by writing 5AF0h to RESET\_KEY of RKR.
10. Read the digital output (CDR[CDATA]) stored in FIFO after the output settling time. Expected output if there is no gain error is 1000\_0110\_0110\_0110b, corresponding to full negative scale after attenuation inserted by the internal filter ( $-1 \times 0.95$ ). The measurement should be repeated to reduce contribution of noise during calibration process.  $D_n$  is the average value of attenuated negative full scale given by  $D_n = \text{AVERAGE}(\text{CDR}[\text{CDATA}])$ .
11. The SDADC calibrated gain can be calculated as:  

$$\text{Gain} = (D_p - D_n) / 2^{16}$$
12. The measured gain value can be used to nullify the gain errors in the digital output. For calibrated conversion, the data CDR[CDATA] provided by the SDADC should be normalized using the calculated gain:  $\text{CDATA}_{\text{norm}}$  is given by  $\text{CDATA}_{\text{norm}} = \text{CDR}[\text{CDATA}] / \text{Gain}$ .

### Note

- a. The gain error calibration should only be done with gain equal to '1' (in other words, PGAN = 000). Nevertheless, the calculated gain can be applied to normalized data also for conversion with  $\text{GAIN} > 1$ .
- b. MCR[GECEN] = 1 ensures the accurate gain error mode is enabled. This bit should be set both during calibration process and after calibration, for application data conversion. Application data conversion must then be normalized using the measured gain.
- c. The higher the number of full-scale conversion ( $D_p$ ,  $D_n$ ) done, the higher the rejection of noise during the calibration. The number of conversion is dependent on application noise. It is recommended to run at least 16 conversions when calculating an average value.

## 28.8.11 Offset calibration support

To perform offset calibration, the following sequence must be applied.

1. Select differential mode of operation by writing SDADC\_MCR[MODE] to '0'.
2. Configure the SDADC\_CSR[ANCHSEL] mux selection field to '100' or '101' as required.
  - a. SDADC\_CSR[ANCHSEL] = '100' in case of data conversion after calibration in "single ended mode with negative input= $V_{SS\_HV\_ADR\_D}$ "
  - b. SDADC\_CSR[ANCHSEL] = '101' in case of data conversion after calibration in "differential mode" and "single ended mode with negative input =  $(V_{DD\_HV\_ADR\_D} - V_{SS\_HV\_ADR\_D}) / 2$ "
3. Disable the bias on all input analog channels by writing ENBIAS field of CSR to 00h.
4. Disable the high pass filter by deasserting SDADC\_MCR[HPFEN].
5. Generate a reset event by writing 5AF0h to RESET\_KEY of RKR.
6. Read the digital output stored in FIFO after the output settling time.
7. The measured offset can be used to nullify the offset error in the digital output.  
Expected output is 00\_0000\_0000\_0000b. The SDADC offset can be calculated as:  
Offset = Expected Output – Actual Output

### Note

The offset must be calculated for each SDADC\_MCR[PGAN] field setting since it is expected to vary with gain configuration of SDADC.

## 28.9 Initialization information

To initialize the SDADC registers for data conversion, the following sequence is required.

1. Enable the SDADC by asserting MCR[EN].
2. Configure MCR to select the required mode, polarity, common mode voltage, input gain, and decimation rate.
3. Enable high-pass filter if required.
4. Select the required analog channel for data conversion. It is possible to select the bias for each channel for AC coupling applications.
5. Configure OSD delay according to SDADC startup time or latency from reset exit.
6. Generate a reset event by writing 5AF0h to RESET\_KEY of RKR. Otherwise, if data conversion need to be triggered by a hardware event, assert MCR[TRIGEN].



7. DFFF interrupt or DMA request will be generated after the data FIFO has reached threshold to indicate that data can be transferred.

### 28.9.1 Data conversion step

To acquire a data from SDADC, the following sequence is required:

1. Enable the SDADC by asserting MCR[EN].
2. Configure MCR to select the required mode, polarity, common mode voltage, input gain, and decimation rate.
3. Enable high-pass filter is required.
4. Select the required analog channel for data conversion. It is possible to select the bias for each channel for AC coupling applications.
5. Configure OSD delay according to SDADC required startup time or latency from reset exit.
6. Generate a reset event writing 5AF0h in RESET\_KEY field of RKR register.
7. Wait till FIFO empty flag DFEF of SFR register is clear.
8. Read data by CDATA of CDR register.

Repeat steps from 6 to 8 for new acquisitions.

#### NOTE

The time elapsed between reset event and Read data by CDATA field of CDR register are:

- Reset event <- -> Data Valid flag(CDVS) = Output settling Time delay (OSDR has to be set at least to 16) (Refer to the  $t_{\text{settling}}$  and  $t_{\text{LATENCY}}$  parameters in the device datasheet for the minimum allowed output settling delay values.).
- Data Valid flag <- -> Data FIFO is not empty =  $0.5 * \text{CLK\_Out} + 3 * \text{CLK\_In}$
- Data FIFO is not empty <- -> read Data (safety point) =  $2 * \text{CLK\_In}$ , where  $\text{CLK\_Out} = \text{CLK\_In} / (2 * \text{OSR})$



## Chapter 29

# Enhanced Queued Analog-to-Digital Converter (EQADC)

### 29.1 Chip-specific EQADC information

This chip has two instances of the EQADC module: EQADC\_A and EQADC\_B.

Each EQADC module:

- controls two independent Redundant Signed Digit (RSD) analog-to-digital (ADC) converter cores: ADC0 and ADC1.
- has an independent digital interface with register interfaces, control hardware, DMA and interrupt requests, and so on.

Chip-specific features of the EQADCs include:

- 24 dedicated external analog input pins to each EQADC module
- 16 pins shared between the two modules
- 6 additional pins share the dedicated EQADC\_B channel inputs for greater pinout flexibility
- No support for external converters or an external device; the SSI interface is not present
- Addition of a Burst Trigger Generator, which is described in the SIU chapter
- Addition of bias/diagnostic pull up/down in the pad cell to support sharing ADC pins with SDADC inputs
- Targets up to 1.5M Sample/s (ADC\_CLK=24 MHz) for differential conversions

The value of the external bypass between capacitor REFBYPC and VRL should be 100 nf.

**NOTE**

Due to legacy reasons, both ADC0 and ADC1 must be enabled for accurate conversion results.

**NOTE**

An 8 ms wait time from VDDA power up to enabling ADC is required to pre-charge the external 100 nf capacitor on REFBYPC pin. This time must be guaranteed by crystal startup time plus reset duration or user. The ADC internal bias generator circuit will start up after 10  $\mu$ s upon VRH/VRL and VDDA/VSSA power up and produces a stable/required bias current to the pre-charge circuit, but the current to other analog circuits are disabled until ADCs are enabled. As soon as the ADCs are enabled, the bias currents to all of analog circuits will be enabled.

### 29.1.1 EQADC connections to other modules

Each EQADC module:

- has internal channel connections with the Power Management Controller module.

#### 29.1.1.1 Sources of software and external hardware triggers

The EQADC supports software and external hardware triggers (via package pins) from the PIT-RTI, eTPU, and eMIOS modules to initiate transfers of commands from the CFIFOs to the on-chip ADCs.

#### 29.1.1.2 STAC server IDs

STAC server IDs are defined in [STAC server IDs](#) in the chip-specific eTPU information.

### 29.1.2 EQADC Stop mode entry

On this chip, SIUL\_HLT1[EQADCx] is the chip-level control for issuing a request to an EQADC instance to enter its Stop mode.

### 29.1.3 EQADC pre-gain control

On this chip, the analog ADC option to select x1, x2, or x4 pre-gain is available only for differential conversions. Single-ended conversions can use only x1 pre-gain.

The PRE\_GAIN field of the [Alternate Configuration 1-14 Control Registers \(ADC\\_ACR1-14\)](#) controls the selection.

### 29.1.4 CFIFO and RFIFO depth

Every CFIFO and RFIFO, including CFIFO0, has four entries.

### 29.1.5 EQADC pin pullup/pulldown control

This following table shows the analog and digital EQADC pins with pullup/pulldown control and the registers that provide the control.

**Table 29-1. EQADC pins with pullup/pulldown and control registers**

| Pin function | EQADC_A                        | EQADC_B                         | Pullup/pulldown control registers                                        |
|--------------|--------------------------------|---------------------------------|--------------------------------------------------------------------------|
| Analog       | ANA0 to ANA7<br>ANA16 to ANA23 | ANB0 to ANB7                    | <a href="#">ADC Pull Up/Down Control Register x (ADC_PUDCRx, x=0-39)</a> |
| Digital      | —                              | ANB8 to ANB23<br>ANB40 to ANB45 | <a href="#">Pad Configuration Register (SIU_PCRn)</a>                    |

### 29.1.6 EQADC pin-to-channel mapping

For both EQADC\_A and EQADC\_B, the following table shows the pin-to-channel mapping to show which signals go to which ADC.

**Table 29-2. EQADC\_A and EQADC\_B pin-to-channel mapping**

| Analog input pin on 416 MAPBGA package | Function                   | ADC number |         | EQADC_A channel number | EQADC_B channel number |
|----------------------------------------|----------------------------|------------|---------|------------------------|------------------------|
|                                        |                            | EQADC_A    | EQADC_B |                        |                        |
| ANA0-ANA23 <sup>1</sup>                | Single ended conversion    | 0, 1       | —       | 0-23                   | —                      |
| ANB0-ANB23                             | Single ended conversion    | —          | 0, 1    | —                      | 0-23                   |
| AN24-AN39                              | Single ended conversion    | 0, 1       | 0, 1    | 24-39                  | 24-39                  |
| —                                      | VRH                        | 0, 1       | 0, 1    | 40                     | 40                     |
| —                                      | VRL                        | 0, 1       | 0, 1    | 41                     | 41                     |
| —                                      | 50% (VRH-VRL) <sup>2</sup> | 0, 1       | 0, 1    | 42                     | 42                     |

*Table continues on the next page...*

**Table 29-2. EQADC\_A and EQADC\_B pin-to-channel mapping (continued)**

| Analog input pin on 416 MAPBGA package | Function                                        | ADC number |         | EQADC_A channel number | EQADC_B channel number |
|----------------------------------------|-------------------------------------------------|------------|---------|------------------------|------------------------|
|                                        |                                                 | EQADC_A    | EQADC_B |                        |                        |
| —                                      | 75% (VRH-VRL)                                   | 0, 1       | 0, 1    | 43                     | 43                     |
| —                                      | 25% (VRH-VRL)                                   | 0, 1       | 0, 1    | 44                     | 44                     |
| —                                      | Local bandgap 1220 mV                           | 0, 1       | —       | 45                     | —                      |
| ANA8                                   | ANWA <sup>3</sup>                               | 0, 1       | —       | 64-71                  | —                      |
| ANB8                                   | ANWB                                            | —          | 0, 1    | —                      | 64-71                  |
| ANA9                                   | ANXA                                            | 0, 1       | —       | 72-79                  | —                      |
| ANB9                                   | ANXB                                            | —          | 0, 1    | —                      | 23-79                  |
| ANA10                                  | ANYA                                            | 0, 1       | —       | 80-87                  | —                      |
| ANB10                                  | ANYB                                            | —          | 0, 1    | —                      | 80-87                  |
| ANA11                                  | ANZA                                            | 0, 1       | —       | 88-95                  | —                      |
| ANB11                                  | ANZB                                            | —          | 0, 1    | —                      | 88-95                  |
| ANA0 and ANA1                          | DANA0+ and DANA0-                               | 0, 1       | —       | 96                     | —                      |
| ANB0 and ANB1                          | DANB0+ and DANB0-                               | —          | 0, 1    | —                      | 96                     |
| ANA2 and ANA3                          | DANA1+ and DANA1-                               | 0, 1       | —       | 97                     | —                      |
| ANB2 and ANB3                          | DANB1+ and DANB1-                               | —          | 0, 1    | —                      | 97                     |
| ANA4 and ANA5                          | DANA2+ and DANA2-                               | 0, 1       | —       | 98                     | —                      |
| ANB4 and ANB5                          | DANB2+ and DANB2-                               | —          | 0, 1    | —                      | 98                     |
| ANA6 and ANA7                          | DANA3+ and DANA3-                               | 0, 1       | —       | 99                     | —                      |
| ANB6 and ANB7                          | DANB3+ and DANB3-                               | —          | 0, 1    | —                      | 99                     |
| —                                      | Temperature Sensor 0                            | 0, 1       | —       | 128                    | —                      |
| —                                      | Temperature Sensor 1                            | 0, 1       | —       | 129                    | —                      |
| —                                      | PMC bandgap 1200 mV                             | 0          | —       | 145 <sup>4</sup>       | —                      |
| —                                      | Core/1.2 V LVD cold sense                       | 0          | —       | 145 <sup>4</sup>       | —                      |
| —                                      | Core/1.2 V LVD hot sense                        | 0          | —       | 145 <sup>4</sup>       | —                      |
| —                                      | Core/1.2 V LVD hot reference                    | 0          | —       | 145 <sup>4</sup>       | —                      |
| —                                      | Flash/3.3 V LVD divider tap                     | 0          | —       | 145 <sup>4</sup>       | —                      |
| —                                      | Flash/3.3 V LVD reference (PMC bandgap 1200 mV) | 0          | —       | 145 <sup>4</sup>       | —                      |
| —                                      | PMC/5 V LVD reference (PMC bandgap 1200 mV)     | 0          | —       | 145 <sup>4</sup>       | —                      |
| ANB40                                  | Single ended conversion                         | —          | 0, 1    | —                      | 128                    |
| ANB41                                  | Single ended conversion                         | —          | 0, 1    | —                      | 129                    |
| ANB42                                  | Single ended conversion                         | —          | 0, 1    | —                      | 130                    |
| ANB43                                  | Single ended conversion                         | —          | 0, 1    | —                      | 131                    |
| ANB44                                  | Single ended conversion                         | —          | 0, 1    | —                      | 132                    |
| ANB45                                  | Single ended conversion                         | —          | 0, 1    | —                      | 133                    |
| ANA16                                  | ANRA                                            | 0, 1       | —       | 224-231                | —                      |
| ANB16                                  | ANRB                                            | —          | 0, 1    | —                      | 245-231                |
| ANA17                                  | ANSA                                            | 0, 1       | —       | 232-239                | —                      |

Table continues on the next page...

**Table 29-2. EQADC\_A and EQADC\_B pin-to-channel mapping (continued)**

| Analog input pin on 416 MAPBGA package | Function | ADC number |         | EQADC_A channel number | EQADC_B channel number |
|----------------------------------------|----------|------------|---------|------------------------|------------------------|
|                                        |          | EQADC_A    | EQADC_B |                        |                        |
| ANB17                                  | ANSB     | —          | 0, 1    | —                      | 223-239                |
| ANA18                                  | ANTA     | 0, 1       | —       | 240-247                | —                      |
| ANB18                                  | ANTB     | —          | 0, 1    | —                      | 240-247                |
| ANA19                                  | ANUA     | 0, 1       | —       | 248-255                | —                      |
| ANB19                                  | ANUB     | —          | 0, 1    | —                      | 248-255                |

1. AN8x-ANx11 and ANx16-ANx19 have alternate functions when used with an external multiplexer.
2. After calibration, the 50% reference point reads approximately 20 mV below 50% (VRH-VRL). The 50% reference point should not be used to calibrate the EQADC. Use the 25% and 75% reference points to calibrate the EQADC.
3. This function is present when using an external multiplexer.
4. Select the desired PMC voltage using the channel select field [ADC\\_CS](#) of the [ADC\\_CS](#)

## 29.2 Introduction

### 29.2.1 Module overview

The Enhanced Queued Analog-to-Digital Converter (EQADC) block provides accurate and fast conversions for a wide range of applications. The EQADC provides a parallel interface to two on-chip analog-to-digital converters (ADCs), and a parallel side interface to one or more on-chip digital signal processing (DSP) modules (for example, a decimation filter). The two on-chip ADCs are architected to allow access to all the analog channels.

The EQADC transfers commands from multiple Command FIFOs (CFIFOs) to the on-chip ADCs. The multiple Result FIFOs (RFIFOs) can receive data from the on-chip ADCs or from an on-chip DSP module. Data from the on-chip ADCs can be routed to the side interface, processed by the on-chip DSP and then routed back through the side interface to the RFIFOs. The EQADC supports software and external hardware triggers from other blocks to initiate transfers of commands from the CFIFOs to the on-chip ADCs. It also monitors the fullness of CFIFOs and RFIFOs, and accordingly generates DMA or interrupt requests to control data movement between the FIFOs and the system memory, which is external to the EQADC.

### 29.2.2 Block diagram

[Figure 29-1](#) is the block diagram for the EQADC.

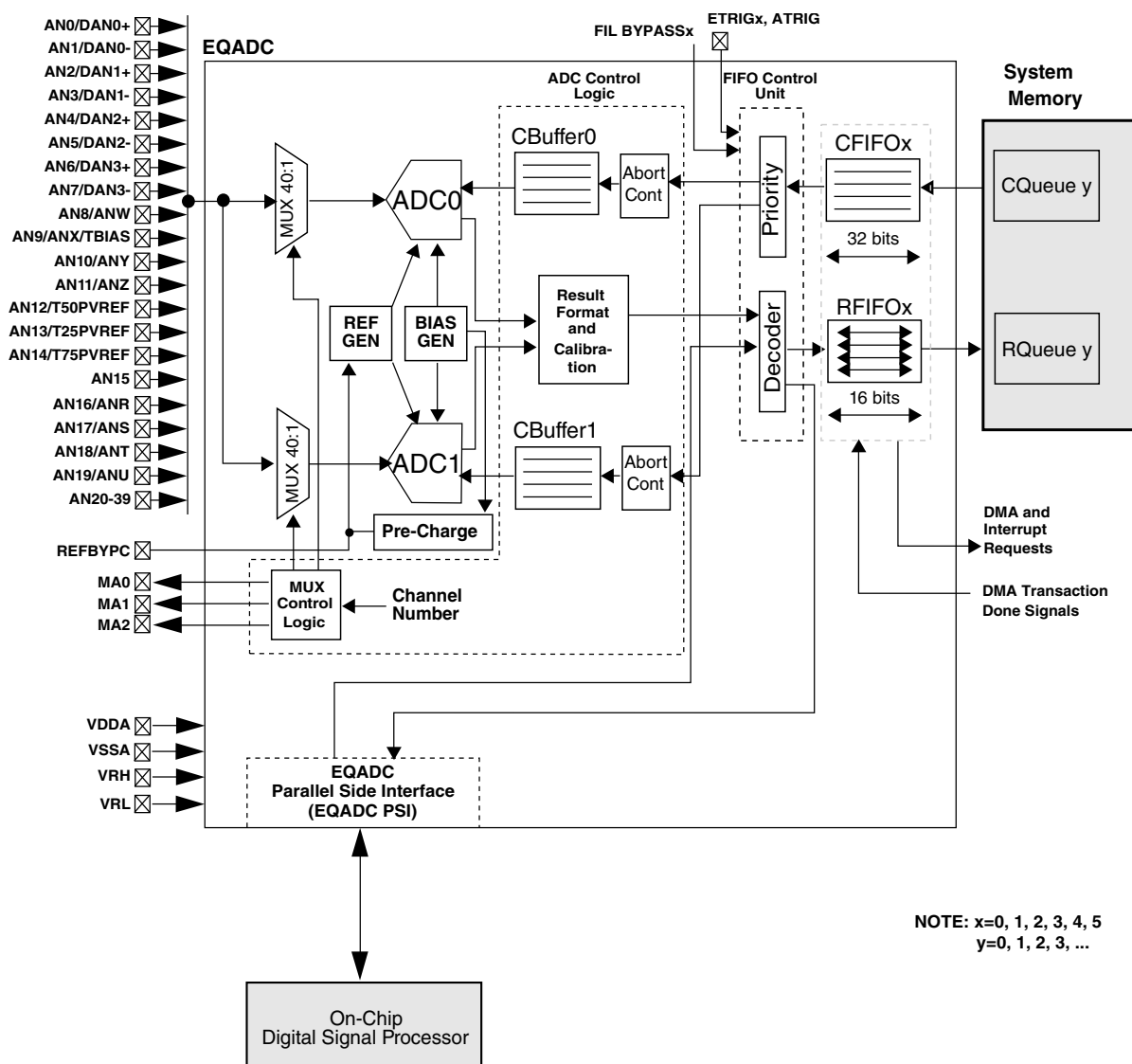


**Figure 29-1. EQADC Block Diagram**

Figure 29-1 shows the primary components inside the EQADC. The EQADC consists of the *FIFO Control Unit* which controls the CFIFOs and the RFIFOs, the *ADC Control Logic* which controls the two on-chip ADCs, and the *EQADC Parallel Side Interface* (EQADC PSI) which allows communication with on-chip eQADC companion<sup>1</sup> modules. There are 6 CFIFOs and 6 RFIFOs, each with 4 entries, except CFIFO0 that can have 8 entries.

The *FIFO Control Unit* performs the following functions:

- It prioritizes the CFIFOs to determine what CFIFOs will have their commands transferred.

1. Depending on the device, companion modules may be DSP modules, decimation filters, or any other processing block.



- Supports software and hardware triggers to start command transfers from a particular CFIFO.
- Decodes command data from the CFIFOs, and accordingly, sends these commands to one of the two on-chip ADCs.
- Decodes result data from on-chip ADCs and transfers data to the appropriate RFIFO or to the parallel side interface.

The *ADC Control Logic* manages the execution of commands bound for on-chip ADCs. It interfaces with the CFIFOs via two 2-entry command buffers (CBuffers) with abort control and with the RFIFOs and side interface via the *Result Format and Calibration Sub-Block*. The *ADC Control Logic* performs the following functions:

- Buffers command data for execution.
- Decodes command data and accordingly generates control signals for the two on-chip ADCs.
- Detects abort request, stores aborted commands and buffers immediate conversion commands.
- Formats and calibrates conversion result data coming from the on-chip ADCs.
- Generates the internal multiplexer control signals and the select signals used by the external multiplexers.

The EQADC PSI allows for a full duplex, synchronous, parallel communication between the EQADC and several on-chip companion modules.

Figure 29-1 also depicts data flow through the EQADC. Commands are contained in system memory in a user defined data structure. The most likely data structure to be used is a queue as depicted in the Figure 29-1<sup>2</sup>. Command data is moved from the command queue (CQueue) to the CFIFOs by either the host CPU or by the DMAC. Once a CFIFO is triggered and becomes the highest priority CFIFO using a certain CBuffer, command data is transferred from the CFIFO to the on-chip ADCs. The ADC executes the command, and the result is moved through the *Result Format and Calibration Sub-Block* to either the side interface or to the RFIFO. Data from the on-chip companion module bypasses the *Result Format and Calibration Sub-Block* and is moved directly to its specified RFIFO. When data is stored in an RFIFO, data is moved from the RFIFO by the host CPU or by the DMAC to a data structure in system memory depicted in the Figure 29-1 as a result queue (RQueue).

2. Command and result data can be stored in system memory in any user defined data structure. However, in this document it will be assumed that the data structure of choice is a queue, since it is the most likely data structure to be used and because queues are the only type of data structure supported by the DMAC.

For users familiar with the QADC, the EQADC system upgrades the functionality provided by that block. See [EQADC versus QADC](#) for a comparison between the EQADC and QADC.

### 29.2.3 Features

The EQADC block includes these distinctive features:

- Two independent on-chip RSD Cyclic ADCs
  - 8, 10, and 12 bits Resolution
  - Differential conversions
  - Differential channels include variable gain amplifier for improved dynamic range (x1; x2; x4)
  - Differential channels include programmable pull-up and pull-down resistors for biasing and sensor diagnostics (200k ohms; 100k ohms; 5k ohms)
  - Sample times of 2 (default), 8, 64 or 128 ADC clock cycles
  - Provides time stamp information when requested
  - Parallel interface to EQADC CFIFOs and RFIFOs
  - Supports both right-justified unsigned and signed formats for conversion results
  - The REFBYPC pin stabilizes one of internal generated reference
  - Temperature sensor
  - Ability to measure directly Vdd
- Automatic application of ADC calibration constants
  - Provision of reference voltages (25%VREF and 75%VREF) for ADC calibration purposes
- 40 input channels available to the two on-chip ADCs
- 4 pairs of differential analog input channels
- Parallel Side Interface to communicate with several on-chip companion modules
- STAC bus to import an alternative timebase to the internal time stamp
- Priority Based CFIFOs

- Supports six CFIFOs with fixed priority. The lower the CFIFO number, the higher its priority. When commands of distinct CFIFOs are bound for the same CBuffer, the higher priority CFIFO is always served first.
- Immediate conversion command feature with conversion abort control
- Streaming mode operation of CFIFO0 to execute some commands several times
- Supports software and several hardware trigger modes to arm a particular CFIFO
- Generates interrupt when command coherency is not achieved
- External Hardware Triggers
  - Supports rising edge, falling edge, high level and low level triggers
  - Supports configurable digital filter
  - Supports controls to bypass the trigger digital filters
- Two Triggers operation mode for queue0
  - Additional internal trigger (not filtered) called Advance trigger that is used to enable the external trigger of queue0 and to control the loop behavior of CFIFO0
- Supports 4 to 8 external 8-to-1 muxes which can expand the input channel number from 40 to 96

## 29.3 Modes of operation

This section describes the operation modes of the EQADC.

### 29.3.1 Normal Mode

This is the default operational mode when the EQADC is not in background debug or stop mode.

### 29.3.2 Debug Mode

Upon a debug mode entry request, EQADC behavior will vary according to the status of the DBG field in [EQADC Module Configuration Register \(EQADC\\_MCR\)](#). If DBG is programmed to 0b00, the debug mode entry request is ignored. If DBG is programmed to 0b10 or to 0b11, the EQADC will enter debug mode.

During debug mode, the EQADC will not transfer commands from any CFIFOs, no null messages will be transmitted to the device, no data will be returned to any RFIFO, no hardware trigger event will be captured, and all EQADC registers can be accessed as in Normal mode. The latter implies that CFIFOs can still be triggered using software triggers, since no scheme is implemented to write-protect registers during debug mode. DMA and interrupt requests continue to be generated as in Normal Mode.

If at the time the debug mode entry request is detected, there are commands in the on-chip CBuffers that were already under execution, these commands will be completed but the generated results, if any, will not be sent to the RFIFOs until debug mode is exited. Commands whose execution has not started will not be executed until debug mode is exited.

The clock associated with an on-chip ADC stops, during its low phase, after the ADC ceases executing commands. The time base counter will only stop after all on-chip ADCs cease executing commands.

When exiting debug mode, the EQADC relies on the CFIFO operation modes and on the CFIFO status to determine the next command entry to transfer.

The EQADC internal behavior after the debug mode entry request is detected differs depending on the status of command transfers.

- No command transfer is in progress.

The EQADC immediately halts future command transfers from any CFIFO.

If a null message is being transmitted, EQADC will complete the serial transmission before halting future command transfers. If valid data (conversion result or data read from an ADC register) is received at the end of transmission, it will not be sent to an RFIFO until debug mode is exited.

If the null message transmission is aborted, the EQADC will complete the abort procedure before halting future command transfers from any CFIFO. The message of the CFIFO that caused the abort of the previous serial transmission will only be transmitted after debug mode is exited.

- Command transfer is in progress.

EQADC will complete the transfer and update CFIFO status before halting future command transfers from any CFIFO. Command transfers to the internal CBuffers are considered completed when a command is written to the buffers.

- Command/Null message transfer through serial interface was aborted but next serial transmission did not start.

If the debug mode entry request is detected between the time a previous serial transmission was aborted and the start of the next transmission, the EQADC will complete the abort procedure before halting future command transfers from any CFIFO. The message of the CFIFO that caused the abort of the previous serial transmission will only be transmitted after debug mode is exited.

### 29.3.3 Stop Mode

Upon a stop mode entry request detection, the EQADC progressively halts its operations until it reaches a static, stable state from which it can recover when returning to Normal mode. EQADC then asserts an acknowledge signal, indicating that it is static and that the clock input can be stopped. . The latter implies that, as long as the system clock is running, CFIFOs can still be triggered using software triggers, since no scheme is implemented to write-protect registers during stop mode.

If at the time the stop mode entry request is detected, there are commands in the on-chip CBuffers that were already under execution, these commands will be completed but the generated results, if any, will not be sent to the RFIFOs until stop mode is exited. Commands whose execution has not started will not be executed until stop mode is exited.

After these remaining commands are executed, the clock input to the ADCs and the bias generator circuit is stopped. The ADC clock stops during its low phase. The time base counter will only stop after all on-chip ADCs cease executing commands. Only then, the stop acknowledge signal is asserted. When exiting stop mode, the EQADC relies on the CFIFO operation modes and on the CFIFO status to determine the next command entry to transfer.

The EQADC internal behavior after the stop mode entry request is detected differs depending on the status of the command transfer.

- No command transfer is in progress

The EQADC immediately halts future command transfers from any CFIFO.

If a null message is being transmitted, EQADC will complete the transmission before halting future command transfers. If valid data (conversion result or data read from an ADC register) is received at the end of the transmission, it will not be sent to an RFIFO until stop mode is exited.

If the null message transmission is aborted, the EQADC will complete the abort procedure before halting future command transfers from any CFIFO. The message of the CFIFO that caused the abort of the previous serial transmission will only be transmitted after stop mode is exited.

- Command transfer is in progress

EQADC will complete the transfer and update CFIFO status before halting future command transfers from any CFIFO. Command transfers to the internal CBuffers are considered completed when a command is written to the buffers.

## 29.4 External signal description

### 29.4.1 Overview

The following is a list of external pins.

#### Note

At chip integration level, some of the digital and analog signals listed here might share pins or not be available to the chip. Refer to the Signals chapter for details.

**Table 29-3. External signals**

| Name      | Port  | Function                                                                | Reset state | Type   |
|-----------|-------|-------------------------------------------------------------------------|-------------|--------|
| AN0/DAN0+ | Input | Single-ended analog input / Differential analog input positive terminal | —           | Analog |
| AN1/DAN0- | Input | Single-ended analog input / Differential analog input negative terminal | —           | Analog |
| AN2/DAN1+ | Input | Single-ended analog input / Differential analog input positive terminal | —           | Analog |
| AN3/DAN1- | Input | Single-ended analog input / Differential analog input negative terminal | —           | Analog |
| AN4/DAN2+ | Input | Single-ended analog input / Differential analog input positive terminal | —           | Analog |
| AN5/DAN2- | Input | Single-ended analog input / Differential analog input negative terminal | —           | Analog |
| AN6/DAN3+ | Input | Single-ended analog input / Differential analog input positive terminal | —           | Analog |
| AN7/DAN3- | Input | Single-ended analog input / Differential analog input negative terminal | —           | Analog |

*Table continues on the next page...*

**Table 29-3. External signals (continued)**

| Name                  | Port   | Function                                                                         | Reset state | Type    |
|-----------------------|--------|----------------------------------------------------------------------------------|-------------|---------|
| AN8/ANW               | Input  | Single-ended analog input / Single-ended analog input from external multiplexers | —           | Analog  |
| AN9/ANX               | Input  | Single-ended analog input / Single-ended analog input from external multiplexers | —           | Analog  |
| AN10/ANY              | Input  | Single-ended analog input/Single-ended analog input from external multiplexers   | —           | Analog  |
| AN11/ANZ              | Input  | Single-ended analog input / Single-ended analog input from external multiplexers | —           | Analog  |
| AN12                  | Input  | Single-ended analog input                                                        | —           | Analog  |
| AN13                  | Input  | Single-ended analog input                                                        | —           | Analog  |
| AN14                  | Input  | Single-ended analog input                                                        | —           | Analog  |
| AN15                  | Input  | Single-ended analog input                                                        | —           | Analog  |
| AN16/ANR <sup>1</sup> | Input  | Single-ended analog input / Single-ended analog input from external multiplexers | —           | Analog  |
| AN17/ANS <sup>1</sup> | Input  | Single-ended analog input / Single-ended analog input from external multiplexers | —           | Analog  |
| AN18/ANT <sup>1</sup> | Input  | Single-ended analog input / Single-ended analog input from external multiplexers | —           | Analog  |
| AN19/ANU <sup>1</sup> | Input  | Single-ended analog input / Single-ended analog input from external multiplexers | —           | Analog  |
| AN20                  | Input  | Single-ended analog input                                                        | —           | Analog  |
| AN21                  | Input  | Single-ended analog input                                                        | —           | Analog  |
| AN22                  | Input  | Single-ended analog input                                                        | —           | Analog  |
| AN23                  | Input  | Single-ended analog input                                                        | —           | Analog  |
| AN24                  | Input  | Single-ended analog input                                                        | —           | Analog  |
| AN25                  | Input  | Single-ended analog input                                                        | —           | Analog  |
| AN26                  | Input  | Single-ended analog input                                                        | —           | Analog  |
| AN27                  | Input  | Single-ended analog input                                                        | —           | Analog  |
| AN28                  | Input  | Single-ended analog input                                                        | —           | Analog  |
| AN29                  | Input  | Single-ended analog input                                                        | —           | Analog  |
| AN30                  | Input  | Single-ended analog input                                                        | —           | Analog  |
| AN31                  | Input  | Single-ended analog input                                                        | —           | Analog  |
| AN32                  | Input  | Single-ended analog input                                                        | —           | Analog  |
| AN33                  | Input  | Single-ended analog input                                                        | —           | Analog  |
| AN34                  | Input  | Single-ended analog input                                                        | —           | Analog  |
| AN35                  | Input  | Single-ended analog input                                                        | —           | Analog  |
| AN36                  | Input  | Single-ended analog input                                                        | —           | Analog  |
| AN37                  | Input  | Single-ended analog input                                                        | —           | Analog  |
| AN38                  | Input  | Single-ended analog input                                                        | —           | Analog  |
| AN39                  | Input  | Single-ended analog input                                                        | —           | Analog  |
| MA0                   | Output | External multiplexer control signal                                              | 0           | Digital |
| MA1                   | Output | External multiplexer control signal                                              | 0           | Digital |

Table continues on the next page...

**Table 29-3. External signals (continued)**

| Name    | Port   | Function                            | Reset state | Type    |
|---------|--------|-------------------------------------|-------------|---------|
| MA2     | Output | External multiplexer control signal | 0           | Digital |
| VDDA    | Input  | Analog Positive Power Supply        | —           | Power   |
| VSSA    | Input  | Analog Negative Power Supply        | —           | Power   |
| VRH     | Input  | Voltage Reference High              | —           | Power   |
| VRL     | Input  | Voltage Reference Low               | —           | Power   |
| REFBYPC | Input  | External Bypass capacitor Pin       | —           | Power   |
| ETRIG0  | Input  | External trigger for CFIFO0         | —           | Digital |
| ETRIG1  | Input  | External trigger for CFIFO1         | —           | Digital |
| ETRIG2  | Input  | External trigger for CFIFO2         | —           | Digital |
| ETRIG3  | Input  | External trigger for CFIFO3         | —           | Digital |
| ETRIG4  | Input  | External trigger for CFIFO4         | —           | Digital |
| ETRIG5  | Input  | External trigger for CFIFO5         | —           | Digital |

1. Can be disabled or not using configuration parameters.

## 29.4.2 Detailed signal descriptions

### 29.4.2.1 AN0/DAN0+ — Single-ended analog input/Differential analog input positive terminal

AN0 is a single-ended analog input to the two on-chip ADCs. DAN0+ is the positive terminal of the differential analog input DAN0 (DAN0+ - DAN0-).

### 29.4.2.2 AN1/DAN0- — Single-ended analog input/Differential analog input negative terminal

AN1 is a single-ended analog input to the two on-chip ADCs. DAN0- is the negative terminal of the differential analog input DAN0 (DAN0+ - DAN0-).

### 29.4.2.3 AN2/DAN1+ — Single-ended analog input/Differential analog input positive terminal

AN2 is a single-ended analog input to the two on-chip ADCs. DAN1+ is the positive terminal of the differential analog input DAN1 (DAN1+ - DAN1-).



#### **29.4.2.4 AN3/DAN1- — Single-ended analog input/Differential analog input negative terminal**

AN3 is a single-ended analog input to the two on-chip ADCs. DAN1- is the negative terminal of the differential analog input DAN1 (DAN1+ - DAN1-).

#### **29.4.2.5 AN4/DAN2+ — Single-ended analog input/Differential analog input positive terminal**

AN4 is a single-ended analog input to the two on-chip ADCs. DAN2+ is the positive terminal of the differential analog input DAN2 (DAN2+ - DAN2-).

#### **29.4.2.6 AN5/DAN2- — Single-ended analog input/Differential analog input negative terminal**

AN5 is a single-ended analog input to the two on-chip ADCs. DAN2- is the negative terminal of the differential analog input DAN2 (DAN2+ - DAN2-).

#### **29.4.2.7 AN6/DAN3+ — Single-ended analog input/Differential analog input positive terminal**

AN6 is a single-ended analog input to the two on-chip ADCs. DAN3+ is the positive terminal of the differential analog input DAN3 (DAN3+ - DAN3-).

#### **29.4.2.8 AN7/DAN3- — Single-ended analog input/Differential analog input negative terminal**

AN7 is a single-ended analog input to the two on-chip ADCs. DAN3- is the negative terminal of the differential analog input DAN3 (DAN3+ - DAN3-).

#### **29.4.2.9 AN8/ANW — Single-ended analog input/ Single-ended analog input from external multiplexers**

AN8 is a single-ended analog input to the two on-chip ADCs. ANW is a single-ended analog input to one of the on-chip ADCs in external multiplexed mode.

#### **29.4.2.10 AN9/ANX/TBIAS — Single-ended analog input/ Single-ended analog input from external multiplexers/Test Bias**

AN9 is a single-ended analog input to the two on-chip ADCs. ANX is a single-ended analog input to one of the on-chip ADCs in external multiplexed mode. TBIAS is used during factory test to verify the bias generator circuit.

#### **29.4.2.11 AN10/ANY — Single-ended analog input/ Single-ended analog input from external multiplexers**

AN10 is a single-ended analog input to the two on-chip ADCs. ANY is a single-ended analog input to one of the on-chip ADCs in external multiplexed mode.

#### **29.4.2.12 AN11/ANZ — Single-ended analog input/ Single-ended analog input from external multiplexers**

AN11 is a single-ended analog input to the two on-chip ADCs. ANZ is a single-ended analog input to one of the on-chip ADCs in external multiplexed mode.

#### **29.4.2.13 AN12/T50PVREF — Single-ended analog input/ Test 50% VREF analog output**

AN12 is a single-ended analog input to the two on-chip ADCs. T50PVREF is used during ADC factory test.

#### **29.4.2.14 AN13/T25PVREF — Single-ended analog input/ Test 25% VREF analog output**

AN13 is a single-ended analog input to the two on-chip ADCs. T25PVREF is used during ADC factory test.

#### **29.4.2.15 AN14/T75PVREF — Single-ended analog input/ Test 75% VREF analog output**

AN14 is a single-ended analog input to the two on-chip ADCs. T75PVREF is used during ADC factory test.

#### **29.4.2.16 AN15 — Single-ended analog input**

AN15 is a single-ended analog inputs to the two on-chip ADCs.

#### **29.4.2.17 AN16/ANR — Single-ended analog input/ Single-ended analog input from external multiplexers**

AN16 is a single-ended analog input to the two on-chip ADCs. ANR is a single-ended analog input to one of the on-chip ADCs in external multiplexed mode. The external multiplexing capability is a device option.

#### **29.4.2.18 AN17/ANS — Single-ended analog input/ Single-ended analog input from external multiplexers**

AN17 is a single-ended analog input to the two on-chip ADCs. ANS is a single-ended analog input to one of the on-chip ADCs in external multiplexed mode. The external multiplexing capability is a device option.

#### **29.4.2.19 AN18/ANT — Single-ended analog input/ Single-ended analog input from external multiplexers**

AN18 is a single-ended analog input to the two on-chip ADCs. ANT is a single-ended analog input to one of the on-chip ADCs in external multiplexed mode. The external multiplexing capability is a device option.

#### **29.4.2.20 AN19/ANU — Single-ended analog input/ Single-ended analog input from external multiplexers**

AN19 is a single-ended analog input to the two on-chip ADCs. ANU is a single-ended analog input to one of the on-chip ADCs in external multiplexed mode. The external multiplexing capability is a device option.

#### **29.4.2.21 AN20-AN39 — Single-ended analog input**

AN20 through AN39 are single-ended analog inputs to the two on-chip ADCs.

**29.4.2.22 INA\_ADC0\_0 - INA\_ADC0\_9 — Single-ended analog input**

INA\_ADC0\_0 through INA\_ADC0\_9 are single-ended analog inputs to the on-chip ADC0.

**29.4.2.23 INA\_ADC1\_0 - INA\_ADC1\_9 — Single-ended analog input**

INA\_ADC1\_0 through INA\_ADC1\_9 are single-ended analog inputs to the on-chip ADC1.

**29.4.2.24 MA0-MA2 — External multiplexer control signals**

MA0, MA1, and MA2 combined form a select signal associated with external multiplexers.

**29.4.2.25 VRH, VRL — Voltage reference high and voltage reference low**

VRH and VRL are voltage references for the ADCs. VRH is the highest voltage reference, while VRL is the lowest voltage reference.

**29.4.2.26 VDDA, VSSA — 5V VDD and VSS for the 5V analog components**

VDDA is the positive power supply pin for the ADCs and VSSA is the negative power supply pin for the ADCs. Refer to electrical specifications.

**29.4.2.27 REFBYPC — Reference Bypass Capacitor**

The REFBYPC pin is used to connect an external bypass capacitor between REFBYPC and VRL. See section "Chip-specific EQADC information" for the capacitor value. This bypass capacitor is used to provide a stable reference voltage for the ADC.

**NOTE**

Prior EQADC versions have had two of these pins. To achieve accurate calibration, more time may be required with only one pin.

### 29.4.2.28 ETRIG0-ETRIG5 — External triggers

The external trigger signals are for hardware triggering. The EQADC can detect rising edge, falling edge, high level and low level on each of the external trigger signals. ETRIGx triggers CFIFOx. The EQADC also supports configurable digital filters for these external trigger signals. These digital filters can be bypassed by individual input control signals called eqadc\_intern\_trig\_selx.

## 29.5 Memory map and register definition

This section provides memory maps and detailed descriptions of all registers. Data written to or read from reserved areas of the memory map is undefined.

This section provides memory maps and detailed descriptions of all registers. Data written to or read from reserved areas of the memory map is undefined.

**EQADC memory map**

| Address offset (hex) | Register name                                                | Width (in bits) | Access | Reset value | Section/ page              |
|----------------------|--------------------------------------------------------------|-----------------|--------|-------------|----------------------------|
| 0                    | EQADC Module Configuration Register (EQADC_MCR)              | 32              | R/W    | 0000_0000h  | <a href="#">29.5.1/981</a> |
| 4                    | EQADC Test Register (EQADC_TST)                              | 32              | R      | 0000_0000h  | <a href="#">29.5.2/982</a> |
| C                    | EQADC External Trigger Digital Filter Register (EQADC_ETDFR) | 32              | R/W    | 0000_0000h  | <a href="#">29.5.3/983</a> |
| 10                   | EQADC CFIFO Push Register (EQADC_CFPR0)                      | 32              | W      | 0000_0000h  | <a href="#">29.5.4/984</a> |
| 14                   | EQADC CFIFO Push Register (EQADC_CFPR1)                      | 32              | W      | 0000_0000h  | <a href="#">29.5.4/984</a> |
| 18                   | EQADC CFIFO Push Register (EQADC_CFPR2)                      | 32              | W      | 0000_0000h  | <a href="#">29.5.4/984</a> |
| 1C                   | EQADC CFIFO Push Register (EQADC_CFPR3)                      | 32              | W      | 0000_0000h  | <a href="#">29.5.4/984</a> |
| 20                   | EQADC CFIFO Push Register (EQADC_CFPR4)                      | 32              | W      | 0000_0000h  | <a href="#">29.5.4/984</a> |
| 24                   | EQADC CFIFO Push Register (EQADC_CFPR5)                      | 32              | W      | 0000_0000h  | <a href="#">29.5.4/984</a> |
| 30                   | EQADC Result FIFO Pop Register (EQADC_RFPR0)                 | 32              | R      | 0000_0000h  | <a href="#">29.5.5/985</a> |
| 34                   | EQADC Result FIFO Pop Register (EQADC_RFPR1)                 | 32              | R      | 0000_0000h  | <a href="#">29.5.5/985</a> |
| 38                   | EQADC Result FIFO Pop Register (EQADC_RFPR2)                 | 32              | R      | 0000_0000h  | <a href="#">29.5.5/985</a> |
| 3C                   | EQADC Result FIFO Pop Register (EQADC_RFPR3)                 | 32              | R      | 0000_0000h  | <a href="#">29.5.5/985</a> |
| 40                   | EQADC Result FIFO Pop Register (EQADC_RFPR4)                 | 32              | R      | 0000_0000h  | <a href="#">29.5.5/985</a> |
| 44                   | EQADC Result FIFO Pop Register (EQADC_RFPR5)                 | 32              | R      | 0000_0000h  | <a href="#">29.5.5/985</a> |
| 50                   | EQADC CFIFO Control Register 0 (EQADC_CFCR0)                 | 32              | R/W    | 0000_0000h  | <a href="#">29.5.6/986</a> |
| 54                   | EQADC CFIFO Control Register 1 (EQADC_CFCR1)                 | 32              | R/W    | 0000_0000h  | <a href="#">29.5.7/990</a> |
| 58                   | EQADC CFIFO Control Register 2 (EQADC_CFCR2)                 | 32              | R/W    | 0000_0000h  | <a href="#">29.5.8/993</a> |

*Table continues on the next page...*

## EQADC memory map (continued)

| Address offset (hex) | Register name                                                | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|--------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 60                   | EQADC Interrupt and DMA Control Register 0 (EQADC_IDCR0)     | 32              | R/W    | 0000_0000h  | <a href="#">29.5.9/996</a>   |
| 64                   | EQADC Interrupt and DMA Control Register 1 (EQADC_IDCR1)     | 32              | R/W    | 0000_0000h  | <a href="#">29.5.10/1000</a> |
| 68                   | EQADC Interrupt and DMA Control Register 2 (EQADC_IDCR2)     | 32              | R/W    | 0000_0000h  | <a href="#">29.5.11/1005</a> |
| 70                   | EQADC FIFO and Interrupt Status Register (EQADC_FISR0)       | 32              | R/W    | 0200_0000h  | <a href="#">29.5.12/1010</a> |
| 74                   | EQADC FIFO and Interrupt Status Register (EQADC_FISR1)       | 32              | R/W    | 0200_0000h  | <a href="#">29.5.12/1010</a> |
| 78                   | EQADC FIFO and Interrupt Status Register (EQADC_FISR2)       | 32              | R/W    | 0200_0000h  | <a href="#">29.5.12/1010</a> |
| 7C                   | EQADC FIFO and Interrupt Status Register (EQADC_FISR3)       | 32              | R/W    | 0200_0000h  | <a href="#">29.5.12/1010</a> |
| 80                   | EQADC FIFO and Interrupt Status Register (EQADC_FISR4)       | 32              | R/W    | 0200_0000h  | <a href="#">29.5.12/1010</a> |
| 84                   | EQADC FIFO and Interrupt Status Register (EQADC_FISR5)       | 32              | R/W    | 0200_0000h  | <a href="#">29.5.12/1010</a> |
| 90                   | EQADC CFIFO Transfer Counter Register 0 (EQADC_CFTCR0)       | 32              | R/W    | 0000_0000h  | <a href="#">29.5.13/1014</a> |
| 94                   | EQADC CFIFO Transfer Counter Register 1 (EQADC_CFTCR1)       | 32              | R/W    | 0000_0000h  | <a href="#">29.5.14/1015</a> |
| 98                   | EQADC CFIFO Transfer Counter Register 2 (EQADC_CFTCR2)       | 32              | R/W    | 0000_0000h  | <a href="#">29.5.15/1016</a> |
| A0                   | EQADC CFIFO Status Snapshot Register 0 (EQADC_CFSSR0)        | 32              | R      | 0000_7800h  | <a href="#">29.5.16/1017</a> |
| A4                   | EQADC CFIFO Status Snapshot Register 1 (EQADC_CFSSR1)        | 32              | R      | 0000_7800h  | <a href="#">29.5.17/1019</a> |
| AC                   | EQADC CFIFO Status Register (EQADC_CFSR)                     | 32              | R      | 0000_0000h  | <a href="#">29.5.18/1021</a> |
| D0                   | EQADC STAC Bus Client Configuration Register (EQADC_STACCCR) | 32              | R/W    | 0000_0000h  | <a href="#">29.5.19/1022</a> |
| 100                  | EQADC CFIFO0 Register (EQADC_CF0R0)                          | 32              | R      | 0000_0000h  | <a href="#">29.5.20/1023</a> |
| 104                  | EQADC CFIFO0 Register (EQADC_CF0R1)                          | 32              | R      | 0000_0000h  | <a href="#">29.5.20/1023</a> |
| 108                  | EQADC CFIFO0 Register (EQADC_CF0R2)                          | 32              | R      | 0000_0000h  | <a href="#">29.5.20/1023</a> |
| 10C                  | EQADC CFIFO0 Register (EQADC_CF0R3)                          | 32              | R      | 0000_0000h  | <a href="#">29.5.20/1023</a> |
| 110                  | EQADC CFIFO0 Extension Register (EQADC_CF0ER0)               | 32              | R      | 0000_0000h  | <a href="#">29.5.21/1024</a> |
| 114                  | EQADC CFIFO0 Extension Register (EQADC_CF0ER1)               | 32              | R      | 0000_0000h  | <a href="#">29.5.21/1024</a> |

Table continues on the next page...

**EQADC memory map (continued)**

| Address offset (hex) | Register name                                  | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 118                  | EQADC CFIFO0 Extension Register (EQADC_CF0ER2) | 32              | R      | 0000_0000h  | <a href="#">29.5.21/1024</a> |
| 11C                  | EQADC CFIFO0 Extension Register (EQADC_CF0ER3) | 32              | R      | 0000_0000h  | <a href="#">29.5.21/1024</a> |
| 140                  | EQADC CFIFO1 Register (EQADC_CF1R0)            | 32              | R      | 0000_0000h  | <a href="#">29.5.22/1024</a> |
| 144                  | EQADC CFIFO1 Register (EQADC_CF1R1)            | 32              | R      | 0000_0000h  | <a href="#">29.5.22/1024</a> |
| 148                  | EQADC CFIFO1 Register (EQADC_CF1R2)            | 32              | R      | 0000_0000h  | <a href="#">29.5.22/1024</a> |
| 14C                  | EQADC CFIFO1 Register (EQADC_CF1R3)            | 32              | R      | 0000_0000h  | <a href="#">29.5.22/1024</a> |
| 180                  | EQADC CFIFO2 Register (EQADC_CF2R0)            | 32              | R      | 0000_0000h  | <a href="#">29.5.23/1025</a> |
| 184                  | EQADC CFIFO2 Register (EQADC_CF2R1)            | 32              | R      | 0000_0000h  | <a href="#">29.5.23/1025</a> |
| 188                  | EQADC CFIFO2 Register (EQADC_CF2R2)            | 32              | R      | 0000_0000h  | <a href="#">29.5.23/1025</a> |
| 18C                  | EQADC CFIFO2 Register (EQADC_CF2R3)            | 32              | R      | 0000_0000h  | <a href="#">29.5.23/1025</a> |
| 1C0                  | EQADC CFIFO3 Register (EQADC_CF3R0)            | 32              | R      | 0000_0000h  | <a href="#">29.5.24/1026</a> |
| 1C4                  | EQADC CFIFO3 Register (EQADC_CF3R1)            | 32              | R      | 0000_0000h  | <a href="#">29.5.24/1026</a> |
| 1C8                  | EQADC CFIFO3 Register (EQADC_CF3R2)            | 32              | R      | 0000_0000h  | <a href="#">29.5.24/1026</a> |
| 1CC                  | EQADC CFIFO3 Register (EQADC_CF3R3)            | 32              | R      | 0000_0000h  | <a href="#">29.5.24/1026</a> |
| 200                  | EQADC CFIFO4 Register (EQADC_CF4R0)            | 32              | R      | 0000_0000h  | <a href="#">29.5.25/1026</a> |
| 204                  | EQADC CFIFO4 Register (EQADC_CF4R1)            | 32              | R      | 0000_0000h  | <a href="#">29.5.25/1026</a> |
| 208                  | EQADC CFIFO4 Register (EQADC_CF4R2)            | 32              | R      | 0000_0000h  | <a href="#">29.5.25/1026</a> |
| 20C                  | EQADC CFIFO4 Register (EQADC_CF4R3)            | 32              | R      | 0000_0000h  | <a href="#">29.5.25/1026</a> |
| 240                  | EQADC CFIFO5 Register (EQADC_CF5R0)            | 32              | R      | 0000_0000h  | <a href="#">29.5.26/1027</a> |
| 244                  | EQADC CFIFO5 Register (EQADC_CF5R1)            | 32              | R      | 0000_0000h  | <a href="#">29.5.26/1027</a> |
| 248                  | EQADC CFIFO5 Register (EQADC_CF5R2)            | 32              | R      | 0000_0000h  | <a href="#">29.5.26/1027</a> |
| 24C                  | EQADC CFIFO5 Register (EQADC_CF5R3)            | 32              | R      | 0000_0000h  | <a href="#">29.5.26/1027</a> |

*Table continues on the next page...*

## EQADC memory map (continued)

| Address offset (hex) | Register name                       | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|-------------------------------------|-----------------|--------|-------------|------------------------------|
| 300                  | EQADC RFIFO0 Register (EQADC_RF0R0) | 32              | R      | 0000_0000h  | <a href="#">29.5.27/1028</a> |
| 304                  | EQADC RFIFO0 Register (EQADC_RF0R1) | 32              | R      | 0000_0000h  | <a href="#">29.5.27/1028</a> |
| 308                  | EQADC RFIFO0 Register (EQADC_RF0R2) | 32              | R      | 0000_0000h  | <a href="#">29.5.27/1028</a> |
| 30C                  | EQADC RFIFO0 Register (EQADC_RF0R3) | 32              | R      | 0000_0000h  | <a href="#">29.5.27/1028</a> |
| 340                  | EQADC RFIFO1 Register (EQADC_RF1R0) | 32              | R      | 0000_0000h  | <a href="#">29.5.28/1028</a> |
| 344                  | EQADC RFIFO1 Register (EQADC_RF1R1) | 32              | R      | 0000_0000h  | <a href="#">29.5.28/1028</a> |
| 348                  | EQADC RFIFO1 Register (EQADC_RF1R2) | 32              | R      | 0000_0000h  | <a href="#">29.5.28/1028</a> |
| 34C                  | EQADC RFIFO1 Register (EQADC_RF1R3) | 32              | R      | 0000_0000h  | <a href="#">29.5.28/1028</a> |
| 380                  | EQADC RFIFO2 Register (EQADC_RF2R0) | 32              | R      | 0000_0000h  | <a href="#">29.5.29/1029</a> |
| 384                  | EQADC RFIFO2 Register (EQADC_RF2R1) | 32              | R      | 0000_0000h  | <a href="#">29.5.29/1029</a> |
| 388                  | EQADC RFIFO2 Register (EQADC_RF2R2) | 32              | R      | 0000_0000h  | <a href="#">29.5.29/1029</a> |
| 38C                  | EQADC RFIFO2 Register (EQADC_RF2R3) | 32              | R      | 0000_0000h  | <a href="#">29.5.29/1029</a> |
| 3C0                  | EQADC RFIFO3 Register (EQADC_RF3R0) | 32              | R      | 0000_0000h  | <a href="#">29.5.30/1030</a> |
| 3C4                  | EQADC RFIFO3 Register (EQADC_RF3R1) | 32              | R      | 0000_0000h  | <a href="#">29.5.30/1030</a> |
| 3C8                  | EQADC RFIFO3 Register (EQADC_RF3R2) | 32              | R      | 0000_0000h  | <a href="#">29.5.30/1030</a> |
| 3CC                  | EQADC RFIFO3 Register (EQADC_RF3R3) | 32              | R      | 0000_0000h  | <a href="#">29.5.30/1030</a> |
| 400                  | EQADC RFIFO4 Register (EQADC_RF4R0) | 32              | R      | 0000_0000h  | <a href="#">29.5.31/1030</a> |
| 404                  | EQADC RFIFO4 Register (EQADC_RF4R1) | 32              | R      | 0000_0000h  | <a href="#">29.5.31/1030</a> |
| 408                  | EQADC RFIFO4 Register (EQADC_RF4R2) | 32              | R      | 0000_0000h  | <a href="#">29.5.31/1030</a> |
| 40C                  | EQADC RFIFO4 Register (EQADC_RF4R3) | 32              | R      | 0000_0000h  | <a href="#">29.5.31/1030</a> |
| 440                  | EQADC RFIFO5 Register (EQADC_RF5R0) | 32              | R      | 0000_0000h  | <a href="#">29.5.32/1031</a> |
| 444                  | EQADC RFIFO5 Register (EQADC_RF5R1) | 32              | R      | 0000_0000h  | <a href="#">29.5.32/1031</a> |

Table continues on the next page...



**EQADC memory map (continued)**

| Address offset (hex) | Register name                       | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|-------------------------------------|-----------------|--------|-------------|------------------------------|
| 448                  | EQADC RFIFO5 Register (EQADC_RF5R2) | 32              | R      | 0000_0000h  | <a href="#">29.5.32/1031</a> |
| 44C                  | EQADC RFIFO5 Register (EQADC_RF5R3) | 32              | R      | 0000_0000h  | <a href="#">29.5.32/1031</a> |

**29.5.1 EQADC Module Configuration Register (EQADC\_MCR)**

The EQADC Module Configuration Register (EQADC\_MCR) contains bits used to control how the EQADC responds to a debug mode entry request.

Address: 0h base + 0h offset = 0h

|       |    |    |    |    |    |    |    |    |       |       |     |    |    |    |     |    |
|-------|----|----|----|----|----|----|----|----|-------|-------|-----|----|----|----|-----|----|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8     | 9     | 10  | 11 | 12 | 13 | 14  | 15 |
| R     | 0  |    |    |    |    |    |    |    |       |       |     |    |    |    |     |    |
| W     |    |    |    |    |    |    |    |    |       |       |     |    |    |    |     |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0     | 0     | 0   | 0  | 0  | 0  | 0   | 0  |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24    | 25    | 26  | 27 | 28 | 29 | 30  | 31 |
| R     | 0  |    |    |    |    |    |    |    | ICEA0 | ICEA1 | DAM | 0  |    | 0  | DBG |    |
| W     |    |    |    |    |    |    |    |    | ICEA0 | ICEA1 | DAM |    |    |    |     |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0     | 0     | 0   | 0  | 0  | 0  | 0   | 0  |

**EQADC\_MCR field descriptions**

| Field            | Description                                                                                                                                                                                                                                                                                            |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–23<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                |
| 24<br>ICEA0      | Immediate Conversion Command Enable ADC0<br>ICEA0 enables the EQADC to abort on-chip ADC0 current conversion and to start the immediate conversion command from CFIFO0 in the requested ADC0.<br><br>0 Disable immediate conversion command request.<br>1 Enable immediate conversion command request. |
| 25<br>ICEA1      | Immediate Conversion Command Enable ADC1<br>ICEA1 enables the EQADC to abort on-chip ADC1 current conversion and to start the immediate conversion command from CFIFO0 in the requested ADC1.<br><br>0 Disable immediate conversion command request.<br>1 Enable immediate conversion command request. |

Table continues on the next page...

## EQADC\_MCR field descriptions (continued)

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 26<br>DAM         | <p>Alternate Configuration Destination Addressing Mode</p> <p>DAM selects between two modes of sending the same conversion result to two different destinations. This is applied mainly when using Alternate Conversion Commands.</p> <p>0 Legacy mode. Uses a defined set of pairs of companion module destinations in the PSI as defined by <a href="#">Table 29-120 in Alternate Configuration 1-14 Control Registers (ADC_ACR1-14)</a>. The Extended Alternate Configuration registers (ADC_EACR1-14) are not considered.</p> <p>1 Selects the Extended Alternate Configuration Registers to send result to two destinations. These destinations can be RFIFOs and companion modules connected to the PSI. See <a href="#">Extended Alternate Configuration 1-14 Control Registers (ADC_EACR1-14)</a> for details.</p> |
| 27–28<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 29<br>Reserved    | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 30–31<br>DBG      | <p>Debug enable</p> <p>The DBG field defines the EQADC response to a debug mode entry request. .</p> <p>0b00 Do not enter debug mode.</p> <p>0b01 Reserved.</p> <p>0b10 Enter debug mode.</p> <p>0b11 Enter debug mode.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

## 29.5.2 EQADC Test Register (EQADC\_TST)

The EQADC Test Register (EQADC\_TST) is used for test purposes only. This register can only be read/written in a test access, accessing the EQADC\_TST register in any other way will result in a transfer error. In a non-test access to the EQADC\_TST register, read data is undefined and written data is ignored.

Address: 0h base + 4h offset = 4h

|       |          |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0        | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | Reserved |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |          |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

## EQADC\_TST field descriptions

| Field            | Description             |
|------------------|-------------------------|
| 0–31<br>Reserved | This field is reserved. |

### 29.5.3 EQADC External Trigger Digital Filter Register (EQADC\_ETDFR)

The EQADC External Trigger Digital Filter Register (EQADC\_ETDFR) is used to set the minimum time a signal must be held in a logic state on the CFIFO triggers inputs to be recognized as an edge or level gated trigger. The Digital Filter Length field specifies the minimum number of system clocks that must be counted by the digital filter counter to recognize a logic state change. However, there is a control signal that can be used to bypass the digital filter when this is not needed.

The count specifies the sample period of the digital filter which is calculated according to the following equation:

$$\text{FilterPeriod} = (\text{SystemClockPeriod} \times 2^{\text{DFL}}) + 1 (\text{SystemClockPeriod})$$

Minimum clock counts for which an ETRIG signal needs to be stable to be passed through the filter are shown in [Table 29-8](#). See [External trigger event detection](#) for more information on the digital filter.

#### NOTE

The DFL field must only be written when the MODEx of all CFIFOs are configured to disabled.

#### NOTE

When the digital filter is bypassed by using the input control, the DFL is not considered and the trigger input signal is not filtered.

Address: 0h base + Ch offset = Ch

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27  | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | DFL |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0  | 0  | 0  |    |

**EQADC\_ETDFR field descriptions**

| Field            | Description                                                                             |
|------------------|-----------------------------------------------------------------------------------------|
| 0–27<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 28–31<br>DFL     | Digital Filter Length                                                                   |

*Table continues on the next page...*

**EQADC\_ETDFR field descriptions (continued)**

| Field | Description                                                                                                                                       |                     |                                               |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|-----------------------------------------------|
|       | The DFL field specifies the minimum number of system clocks that must be counted by the digital filter counter to recognize a logic state change. |                     |                                               |
|       | DFL[ 0:3]                                                                                                                                         | Minimum Clock Count | Minimum Time (ns)<br>(system clock = 120 MHz) |
|       | 0b0000                                                                                                                                            | 2                   | 16.66                                         |
|       | 0b0001                                                                                                                                            | 3                   | 25.00                                         |
|       | 0b0010                                                                                                                                            | 5                   | 41.66                                         |
|       | 0b0011                                                                                                                                            | 9                   | 75.00                                         |
|       | 0b0100                                                                                                                                            | 17                  | 141.66                                        |
|       | 0b0101                                                                                                                                            | 33                  | 275.00                                        |
|       | 0b0110                                                                                                                                            | 65                  | 541.66                                        |
|       | 0b0111                                                                                                                                            | 129                 | 1075.00                                       |
|       | 0b1000                                                                                                                                            | 257                 | 2141.66                                       |
|       | 0b1001                                                                                                                                            | 513                 | 4275.00                                       |
|       | 0b1010                                                                                                                                            | 1025                | 8541.66                                       |
|       | 0b1011                                                                                                                                            | 2049                | 17075.00                                      |
|       | 0b1100                                                                                                                                            | 4097                | 34141.00                                      |
|       | 0b1101                                                                                                                                            | 8193                | 68275.00                                      |
|       | 0b1110                                                                                                                                            | 16385               | 136541.66                                     |
|       | 0b1111                                                                                                                                            | 32769               | 273075.00                                     |

**29.5.4 EQADC CFIFO Push Register (EQADC\_CFPRn)**

The EQADC CFIFO Push Registers (EQADC\_CFPR) provide a mechanism to fill the CFIFOs with command messages from the CQueues. Refer to [EQADC command FIFOs](#) for more information on the CFIFOs and to [Message format in EQADC](#) for a description on command message formats.

**NOTE**

Only whole words must be written to EQADC\_CFPR. Writing half-words or bytes to EQADC\_CFPR will still push the whole 32-bit CF\_PUSH field into the corresponding CFIFO, but undefined data will fill the areas of CF\_PUSH that were not specifically designated as target locations for the write.

Address: 0h base + 10h offset + (4d × i), where i=0d to 5d

|       |          |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0        | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0        |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     | CF_PUSHX |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

### EQADC\_CFPR<sub>n</sub> field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>CF_PUSHX | <p>CFIFO Push Data x</p> <p>When CFIFO<sub>x</sub> is not full, writing to the whole word or any bytes of EQADC_CFPR<sub>x</sub> will push the 32-bit CF_PUSH<sub>x</sub> value into CFIFO<sub>x</sub>. Writing to the CF_PUSH<sub>x</sub> field also increments the corresponding CFCTR<sub>x</sub> value by one in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISR<sub>n</sub>)</a> . When the CFIFO<sub>x</sub> is full, the EQADC ignores any write to the CF_PUSH<sub>x</sub>. Reading the EQADC_CFPR<sub>x</sub> always returns zero.</p> |

## 29.5.5 EQADC Result FIFO Pop Register (EQADC\_RFPR<sub>n</sub>)

The EQADC Result FIFO Pop Registers (EQADC\_RFPR) provide a mechanism to retrieve data from RFIFOs.

Address: 0h base + 30h offset + (4d × i), where i=0d to 5d

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|---------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16      | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | RF_POPX |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0       | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

### EQADC\_RFPR<sub>n</sub> field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 16–31<br>RF_POPX | <p>Result FIFO Pop Data x</p> <p>When RFIFO<sub>x</sub> is not empty, the RF_POP<sub>x</sub> contains the next unread entry value of RFIFO<sub>x</sub>. Reading a word, a half-word, or any bytes from EQADC_RFPR<sub>x</sub> will pop one entry from RFIFO<sub>x</sub> and cause the RFCTR<sub>x</sub> field to be decremented by one in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISR<sub>n</sub>)</a> . When the RFIFO<sub>x</sub> is empty, any read on EQADC_RFPR<sub>x</sub> returns undefined data value and does not decrement the RFCTR<sub>x</sub> value. Writing to EQADC_RFPR<sub>x</sub> has no effect.</p> |

## 29.5.6 EQADC CFIFO Control Register 0 (EQADC\_CFCR0)

The EQADC CFIFO Control Registers (EQADC\_CFCR) contain bits that affect CFIFOs. These bits specify the CFIFO operation mode and can invalidate all of the CFIFO contents.

Address: 0h base + 50h offset = 50h

|       |    |    |    |        |        |      |        |    |       |    |    |    |        |    |    |    |
|-------|----|----|----|--------|--------|------|--------|----|-------|----|----|----|--------|----|----|----|
| Bit   | 0  | 1  | 2  | 3      | 4      | 5    | 6      | 7  | 8     | 9  | 10 | 11 | 12     | 13 | 14 | 15 |
| R     | 0  |    |    | CFEEE0 | STRME0 | SSE0 | CFINV0 | 0  | MODE0 |    |    |    | AMODE0 |    |    |    |
| W     |    |    |    |        |        |      |        |    |       |    |    |    |        |    |    |    |
| Reset | 0  | 0  | 0  | 0      | 0      | 0    | 0      | 0  | 0     | 0  | 0  | 0  | 0      | 0  | 0  | 0  |
| Bit   | 16 | 17 | 18 | 19     | 20     | 21   | 22     | 23 | 24    | 25 | 26 | 27 | 28     | 29 | 30 | 31 |
| R     | 0  |    |    |        |        |      | CFINV1 | 0  | MODE1 |    |    |    | 0      |    |    |    |
| W     |    |    |    |        |        |      |        |    |       |    |    |    |        |    |    |    |
| Reset | 0  | 0  | 0  | 0      | 0      | 0    | 0      | 0  | 0     | 0  | 0  | 0  | 0      | 0  | 0  | 0  |

### EQADC\_CFCR0 field descriptions

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–2<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                |
| 3<br>CFEEE0     | CFIFO0 Entry Number Extension Enable<br>The CFEEE0 bit is used to enable the extension of the CFIFO0 entries. When in extended mode, the CFIFO0 total entries is the sum of normal entries plus the defined value for extension. For more details, see <a href="#">CFIFO0 streaming mode description</a> .<br><br>0 CFIFO0 has a normal value of entries.<br>1 Enable the extension of CFIFO0 entries. |
| 4<br>STRME0     | CFIFO0 Streaming Mode Operation Enable<br>The STRME0 bit is used to enable the streaming mode of operation of CFIFO0. In this case, it is possible to repeat some sequence of commands of this FIFO. For more details, refer to <a href="#">CFIFO0 streaming mode description</a> .<br><br>0 Streaming mode of CFIFO0 is disabled.<br>1 Enable the streaming mode of CFIFO0.                           |
| 5<br>SSE0       | CFIFO Single-Scan Enable Bit 0                                                                                                                                                                                                                                                                                                                                                                         |

Table continues on the next page...

## EQADC\_CFCR0 field descriptions (continued)

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                 | <p>The SSEx bit is used to set the SSSx bit in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> . Writing a "1" to SSEx will set the SSSx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a>, if the CFIFO is in single-scan mode. When SSSx is already asserted, writing a "1" to SSEx also has no effect. If the CFIFO is in continuous-scan mode or is disabled, writing a "1" to SSEx will not set SSSx. Writing a "0" to SSEx has no effect. SSEx always is read as "0".</p> <p>0 No effect.<br/>1 Set the SSSx bit.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 6<br>CFINV0     | <p>CFIFO Invalidate Bit 0</p> <p>The CFINVx bit causes the EQADC to invalidate all entries of CFIFOx. Writing a "1" to CFINVx will reset the value of CFCTRx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> . Writing a "1" to CFINVx also resets the Push Next Data Pointer, Transfer Next Data Pointer to the first entry of CFIFOx in <a href="#">Figure 29-123</a>. CFINVx always is read as "0". Writing a "0" has no effect.</p> <p><b>NOTE:</b> Writing CFINVx only invalidates commands stored in CFIFOx; previously transferred commands that are waiting for execution, that is commands stored in the CBuffers, will still be executed, and results generated by them will be stored in the appropriate RFIFO.</p> <p><b>NOTE:</b> CFINVx must not be written unless the MODEx is configured to disabled, and CFIFO status is IDLE.</p> <p>0 No effect.<br/>1 Invalidate all of the entries in the corresponding CFIFO.</p>                                                                                                                                                                                       |
| 7<br>Reserved   | <p>This field is reserved.<br/>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 8–11<br>MODE0   | <p>CFIFO Operation Mode 0</p> <p>The MODEx field selects the CFIFO operation mode for CFIFOx. See <a href="#">CFIFO scan trigger modes</a> for more information on CFIFO trigger mode.</p> <p><b>NOTE:</b> If MODEx is not disabled, it must not be changed to any other mode besides disabled. If MODEx is disabled and the CFIFO status is IDLE, MODEx can be changed to any other mode.</p> <p>0000 Disabled<br/>0001 Software Trigger, Single Scan<br/>0010 Low Level Gated External Trigger, Single Scan<br/>0011 High Level Gated External Trigger, Single Scan<br/>0100 Falling Edge External Trigger, Single Scan<br/>0101 Rising Edge External Trigger, Single Scan<br/>0110 Falling or Rising Edge External Trigger, Single Scan<br/>0111 Reserved<br/>1000 Reserved<br/>1001 Software Trigger, Continuous Scan<br/>1010 Low Level Gated External Trigger, Continuous Scan<br/>1011 High Level Gated External Trigger, Continuous Scan<br/>1100 Falling Edge External Trigger, Continuous Scan<br/>1101 Rising Edge External Trigger, Continuous Scan<br/>1110 Falling or Rising Edge External Trigger, Continuous Scan<br/>1111 Reserved</p> |
| 12–15<br>AMODE0 | <p>CFIFO0 Advance Trigger Operation Mode 0</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

Table continues on the next page...

## EQADC\_CFCR0 field descriptions (continued)

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | <p>The AMODE0 field selects the trigger mode for the ATRIG0 trigger signal in streaming mode. The use of reserved values drives to unknown behavior of the block.</p> <p><b>NOTE:</b> If AMODE0 is not disabled, it must not be changed to any other mode besides disabled. If AMODE0 is disabled and the CFIFO0 status is IDLE, AMODE0 can be changed to any other mode.</p> <p><b>NOTE:</b> For the streaming mode of operation when the ATRIG0 is used to enable the ETRIG0 or to advance the command queue, the normal mode of operation is external trigger single scan. Other settings are not fully tested.</p> <p>0000 Disabled</p> <p>0001 Software Trigger, Single Scan</p> <p>0010 Low Level Gated External Trigger, Single Scan</p> <p>0011 High Level Gated External Trigger, Single Scan</p> <p>0100 Falling Edge External Trigger, Single Scan</p> <p>0101 Rising Edge External Trigger, Single Scan</p> <p>0110 Falling or Rising Edge External Trigger, Single Scan</p> <p>0111 Reserved</p> <p>1000 Reserved</p> <p>1001 Reserved</p> <p>1010 Reserved</p> <p>1011 Reserved</p> <p>1100 Reserved</p> <p>1101 Reserved</p> <p>1110 Reserved</p> <p>1111 Reserved</p> |
| 16–20<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 21<br>SSE1        | <p>CFIFO Single-Scan Enable Bit 1</p> <p>The SSEx bit is used to set the SSSx bit in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a>. Writing a "1" to SSEx will set the SSSx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a>, if the CFIFO is in single-scan mode. When SSSx is already asserted, writing a "1" to SSEx also has no effect. If the CFIFO is in continuous-scan mode or is disabled, writing a "1" to SSEx will not set SSSx. Writing a "0" to SSEx has no effect. SSEx always is read as "0".</p> <p>0 No effect.</p> <p>1 Set the SSSx bit.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 22<br>CFINV1      | <p>CFIFO Invalidate Bit 1</p> <p>The CFINVx bit causes the EQADC to invalidate all entries of CFIFOx. Writing a "1" to CFINVx will reset the value of CFCTRx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a>. Writing a "1" to CFINVx also resets the Push Next Data Pointer, Transfer Next Data Pointer to the first entry of CFIFOx in <a href="#">Figure 29-123</a>. CFINVx always is read as "0". Writing a "0" has no effect.</p> <p><b>NOTE:</b> Writing CFINVx only invalidates commands stored in CFIFOx; previously transferred commands that are waiting for execution, that is commands stored in the CBuffers, will still be executed, and results generated by them will be stored in the appropriate RFIFO.</p> <p><b>NOTE:</b> CFINVx must not be written unless the MODEx is configured to disabled, and CFIFO status is IDLE.</p>                                                                                                                                                                                                                                                                                                          |

Table continues on the next page...



**EQADC\_CFCR0 field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | 0 No effect.<br>1 Invalidate all of the entries in the corresponding CFIFO.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 23<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 24–27<br>MODE1    | <p>CFIFO Operation Mode 1</p> <p>The MODEx field selects the CFIFO operation mode for CFIFOx. See <a href="#">CFIFO scan trigger modes</a> for more information on CFIFO trigger mode.</p> <p><b>NOTE:</b> If MODEx is not disabled, it must not be changed to any other mode besides disabled. If MODEx is disabled and the CFIFO status is IDLE, MODEx can be changed to any other mode.</p> <p>0000 Disabled<br/> 0001 Software Trigger, Single Scan<br/> 0010 Low Level Gated External Trigger, Single Scan<br/> 0011 High Level Gated External Trigger, Single Scan<br/> 0100 Falling Edge External Trigger, Single Scan<br/> 0101 Rising Edge External Trigger, Single Scan<br/> 0110 Falling or Rising Edge External Trigger, Single Scan<br/> 0111 Reserved<br/> 1000 Reserved<br/> 1001 Software Trigger, Continuous Scan<br/> 1010 Low Level Gated External Trigger, Continuous Scan<br/> 1011 High Level Gated External Trigger, Continuous Scan<br/> 1100 Falling Edge External Trigger, Continuous Scan<br/> 1101 Rising Edge External Trigger, Continuous Scan<br/> 1110 Falling or Rising Edge External Trigger, Continuous Scan<br/> 1111 Reserved</p> |
| 28–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

## 29.5.7 EQADC CFIFO Control Register 1 (EQADC\_CFCR1)

The EQADC CFIFO Control Registers (EQADC\_CFCR) contain bits that affect CFIFOs. These bits specify the CFIFO operation mode and can invalidate all of the CFIFO contents.

Address: 0h base + 54h offset = 54h

|       |   |   |   |   |   |      |        |   |       |   |    |    |    |    |    |    |  |
|-------|---|---|---|---|---|------|--------|---|-------|---|----|----|----|----|----|----|--|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5    | 6      | 7 | 8     | 9 | 10 | 11 | 12 | 13 | 14 | 15 |  |
| R     | 0 |   |   |   |   | SSE2 | CFINV2 | 0 | MODE2 |   |    |    |    | 0  |    |    |  |
| W     |   |   |   |   |   |      |        |   |       |   |    |    |    |    |    |    |  |
| Reset | 0 | 0 | 0 | 0 | 0 | 0    | 0      | 0 | 0     | 0 | 0  | 0  | 0  | 0  | 0  | 0  |  |

|       |    |    |    |    |    |      |        |    |       |    |    |    |    |    |    |    |  |
|-------|----|----|----|----|----|------|--------|----|-------|----|----|----|----|----|----|----|--|
| Bit   | 16 | 17 | 18 | 19 | 20 | 21   | 22     | 23 | 24    | 25 | 26 | 27 | 28 | 29 | 30 | 31 |  |
| R     | 0  |    |    |    |    | SSE3 | CFINV3 | 0  | MODE3 |    |    |    |    | 0  |    |    |  |
| W     |    |    |    |    |    |      |        |    |       |    |    |    |    |    |    |    |  |
| Reset | 0  | 0  | 0  | 0  | 0  | 0    | 0      | 0  | 0     | 0  | 0  | 0  | 0  | 0  | 0  | 0  |  |

### EQADC\_CFCR1 field descriptions

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–4<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 5<br>SSE2       | <p>CFIFO Single-Scan Enable Bit 2</p> <p>The SSE<sub>x</sub> bit is used to set the SSS<sub>x</sub> bit in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISR<sub>n</sub>)</a>. Writing a "1" to SSE<sub>x</sub> will set the SSS<sub>x</sub> in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISR<sub>n</sub>)</a>, if the CFIFO is in single-scan mode. When SSS<sub>x</sub> is already asserted, writing a "1" to SSE<sub>x</sub> also has no effect. If the CFIFO is in continuous-scan mode or is disabled, writing a "1" to SSE<sub>x</sub> will not set SSS<sub>x</sub>. Writing a "0" to SSE<sub>x</sub> has no effect. SSE<sub>x</sub> always is read as "0".</p> <p>0 No effect.<br/>1 Set the SSS<sub>x</sub> bit.</p>                                                                                                                                                                                                                                                   |
| 6<br>CFINV2     | <p>CFIFO Invalidate Bit 2</p> <p>The CFINV<sub>x</sub> bit causes the EQADC to invalidate all entries of CFIFO<sub>x</sub>. Writing a "1" to CFINV<sub>x</sub> will reset the value of CFCTR<sub>x</sub> in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISR<sub>n</sub>)</a>. Writing a "1" to CFINV<sub>x</sub> also resets the Push Next Data Pointer, Transfer Next Data Pointer to the first entry of CFIFO<sub>x</sub> in <a href="#">Figure 29-123</a>. CFINV<sub>x</sub> always is read as "0". Writing a "0" has no effect.</p> <p><b>NOTE:</b> Writing CFINV<sub>x</sub> only invalidates commands stored in CFIFO<sub>x</sub>; previously transferred commands that are waiting for execution, that is commands stored in the CBuffers, will still be executed, and results generated by them will be stored in the appropriate RFIFO.</p> <p><b>NOTE:</b> CFINV<sub>x</sub> must not be written unless the MODE<sub>x</sub> is configured to disabled, and CFIFO status is IDLE.</p> |

Table continues on the next page...

## EQADC\_CFCR1 field descriptions (continued)

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | 0 No effect.<br>1 Invalidate all of the entries in the corresponding CFIFO.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 7<br>Reserved     | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 8–11<br>MODE2     | CFIFO Operation Mode 2<br>The MODEx field selects the CFIFO operation mode for CFIFOx. See <a href="#">CFIFO scan trigger modes</a> for more information on CFIFO trigger mode.<br><br><b>NOTE:</b> If MODEx is not disabled, it must not be changed to any other mode besides disabled. If MODEx is disabled and the CFIFO status is IDLE, MODEx can be changed to any other mode.<br><br>0000 Disabled<br>0001 Software Trigger, Single Scan<br>0010 Low Level Gated External Trigger, Single Scan<br>0011 High Level Gated External Trigger, Single Scan<br>0100 Falling Edge External Trigger, Single Scan<br>0101 Rising Edge External Trigger, Single Scan<br>0110 Falling or Rising Edge External Trigger, Single Scan<br>0111 Reserved<br>1000 Reserved<br>1001 Software Trigger, Continuous Scan<br>1010 Low Level Gated External Trigger, Continuous Scan<br>1011 High Level Gated External Trigger, Continuous Scan<br>1100 Falling Edge External Trigger, Continuous Scan<br>1101 Rising Edge External Trigger, Continuous Scan<br>1110 Falling or Rising Edge External Trigger, Continuous Scan<br>1111 Reserved |
| 12–20<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 21<br>SSE3        | CFIFO Single-Scan Enable Bit 3<br>The SSEx bit is used to set the SSSx bit in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> . Writing a "1" to SSEx will set the SSSx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> , if the CFIFO is in single-scan mode. When SSSx is already asserted, writing a "1" to SSEx also has no effect. If the CFIFO is in continuous-scan mode or is disabled, writing a "1" to SSEx will not set SSSx. Writing a "0" to SSEx has no effect. SSEx always is read as "0".<br><br>0 No effect.<br>1 Set the SSSx bit.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 22<br>CFINV3      | CFIFO Invalidate Bit 3<br>The CFINVx bit causes the EQADC to invalidate all entries of CFIFOx. Writing a "1" to CFINVx will reset the value of CFCTRx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> . Writing a "1" to CFINVx also resets the Push Next Data Pointer, Transfer Next Data Pointer to the first entry of CFIFOx in <a href="#">Figure 29-123</a> . CFINVx always is read as "0". Writing a "0" has no effect.<br><br><b>NOTE:</b> Writing CFINVx only invalidates commands stored in CFIFOx; previously transferred commands that are waiting for execution, that is commands stored in the CBuffers, will still be executed, and results generated by them will be stored in the appropriate RFIFO.<br><br><b>NOTE:</b> CFINVx must not be written unless the MODEx is configured to disabled, and CFIFO status is IDLE.                                                                                                                                                                                                                                                           |

Table continues on the next page...

**EQADC\_CFCR1 field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | 0 No effect.<br>1 Invalidate all of the entries in the corresponding CFIFO.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 23<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 24–27<br>MODE3    | <p>CFIFO Operation Mode 3</p> <p>The MODEx field selects the CFIFO operation mode for CFIFOx. See <a href="#">CFIFO scan trigger modes</a> for more information on CFIFO trigger mode.</p> <p><b>NOTE:</b> If MODEx is not disabled, it must not be changed to any other mode besides disabled. If MODEx is disabled and the CFIFO status is IDLE, MODEx can be changed to any other mode.</p> <p>0000 Disabled<br/> 0001 Software Trigger, Single Scan<br/> 0010 Low Level Gated External Trigger, Single Scan<br/> 0011 High Level Gated External Trigger, Single Scan<br/> 0100 Falling Edge External Trigger, Single Scan<br/> 0101 Rising Edge External Trigger, Single Scan<br/> 0110 Falling or Rising Edge External Trigger, Single Scan<br/> 0111 Reserved<br/> 1000 Reserved<br/> 1001 Software Trigger, Continuous Scan<br/> 1010 Low Level Gated External Trigger, Continuous Scan<br/> 1011 High Level Gated External Trigger, Continuous Scan<br/> 1100 Falling Edge External Trigger, Continuous Scan<br/> 1101 Rising Edge External Trigger, Continuous Scan<br/> 1110 Falling or Rising Edge External Trigger, Continuous Scan<br/> 1111 Reserved</p> |
| 28–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

## 29.5.8 EQADC CFIFO Control Register 2 (EQADC\_CFCR2)

The EQADC CFIFO Control Registers (EQADC\_CFCR) contain bits that affect CFIFOs. These bits specify the CFIFO operation mode and can invalidate all of the CFIFO contents.

Address: 0h base + 58h offset = 58h

| Bit   | 0 | 1 | 2 | 3 | 4 | 5    | 6      | 7 | 8     | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|-------|---|---|---|---|---|------|--------|---|-------|---|----|----|----|----|----|----|
| R     | 0 |   |   |   |   | SSE4 | CFINV4 | 0 | MODE4 |   |    |    |    | 0  |    |    |
| W     |   |   |   |   |   |      |        |   |       |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0    | 0      | 0 | 0     | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

| Bit   | 16 | 17 | 18 | 19 | 20 | 21   | 22     | 23 | 24    | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|-------|----|----|----|----|----|------|--------|----|-------|----|----|----|----|----|----|----|
| R     | 0  |    |    |    |    | SSE5 | CFINV5 | 0  | MODE5 |    |    |    |    | 0  |    |    |
| W     |    |    |    |    |    |      |        |    |       |    |    |    |    |    |    |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0    | 0      | 0  | 0     | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

### EQADC\_CFCR2 field descriptions

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–4<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 5<br>SSE4       | CFIFO Single-Scan Enable Bit 4<br>The SSE <sub>x</sub> bit is used to set the SSS <sub>x</sub> bit in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISR<sub>n</sub>)</a> . Writing a "1" to SSE <sub>x</sub> will set the SSS <sub>x</sub> in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISR<sub>n</sub>)</a> , if the CFIFO is in single-scan mode. When SSS <sub>x</sub> is already asserted, writing a "1" to SSE <sub>x</sub> also has no effect. If the CFIFO is in continuous-scan mode or is disabled, writing a "1" to SSE <sub>x</sub> will not set SSS <sub>x</sub> . Writing a "0" to SSE <sub>x</sub> has no effect. SSE <sub>x</sub> always is read as "0".<br><br>0 No effect.<br>1 Set the SSS <sub>x</sub> bit.                                                                                                                                                                                                                                                     |
| 6<br>CFINV4     | CFIFO Invalidate Bit 4<br>The CFINV <sub>x</sub> bit causes the EQADC to invalidate all entries of CFIFO <sub>x</sub> . Writing a "1" to CFINV <sub>x</sub> will reset the value of CFCTR <sub>x</sub> in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISR<sub>n</sub>)</a> . Writing a "1" to CFINV <sub>x</sub> also resets the Push Next Data Pointer, Transfer Next Data Pointer to the first entry of CFIFO <sub>x</sub> in <a href="#">Figure 29-123</a> . CFINV <sub>x</sub> always is read as "0". Writing a "0" has no effect.<br><br><b>NOTE:</b> Writing CFINV <sub>x</sub> only invalidates commands stored in CFIFO <sub>x</sub> ; previously transferred commands that are waiting for execution, that is commands stored in the CBuffers, will still be executed, and results generated by them will be stored in the appropriate RFIFO.<br><br><b>NOTE:</b> CFINV <sub>x</sub> must not be written unless the MODE <sub>x</sub> is configured to disabled, and CFIFO status is IDLE. |

Table continues on the next page...

## EQADC\_CFCR2 field descriptions (continued)

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | 0 No effect.<br>1 Invalidate all of the entries in the corresponding CFIFO.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 7<br>Reserved     | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 8–11<br>MODE4     | CFIFO Operation Mode 4<br>The MODEx field selects the CFIFO operation mode for CFIFOx. See <a href="#">CFIFO scan trigger modes</a> for more information on CFIFO trigger mode.<br><br><b>NOTE:</b> If MODEx is not disabled, it must not be changed to any other mode besides disabled. If MODEx is disabled and the CFIFO status is IDLE, MODEx can be changed to any other mode.<br><br>0000 Disabled<br>0001 Software Trigger, Single Scan<br>0010 Low Level Gated External Trigger, Single Scan<br>0011 High Level Gated External Trigger, Single Scan<br>0100 Falling Edge External Trigger, Single Scan<br>0101 Rising Edge External Trigger, Single Scan<br>0110 Falling or Rising Edge External Trigger, Single Scan<br>0111 Reserved<br>1000 Reserved<br>1001 Software Trigger, Continuous Scan<br>1010 Low Level Gated External Trigger, Continuous Scan<br>1011 High Level Gated External Trigger, Continuous Scan<br>1100 Falling Edge External Trigger, Continuous Scan<br>1101 Rising Edge External Trigger, Continuous Scan<br>1110 Falling or Rising Edge External Trigger, Continuous Scan<br>1111 Reserved |
| 12–20<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 21<br>SSE5        | CFIFO Single-Scan Enable Bit 5<br>The SSEx bit is used to set the SSSx bit in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> . Writing a "1" to SSEx will set the SSSx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> , if the CFIFO is in single-scan mode. When SSSx is already asserted, writing a "1" to SSEx also has no effect. If the CFIFO is in continuous-scan mode or is disabled, writing a "1" to SSEx will not set SSSx. Writing a "0" to SSEx has no effect. SSEx always is read as "0".<br><br>0 No effect.<br>1 Set the SSSx bit.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 22<br>CFINV5      | CFIFO Invalidate Bit 5<br>The CFINVx bit causes the EQADC to invalidate all entries of CFIFOx. Writing a "1" to CFINVx will reset the value of CFCTRx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> . Writing a "1" to CFINVx also resets the Push Next Data Pointer, Transfer Next Data Pointer to the first entry of CFIFOx in <a href="#">Figure 29-123</a> . CFINVx always is read as "0". Writing a "0" has no effect.<br><br><b>NOTE:</b> Writing CFINVx only invalidates commands stored in CFIFOx; previously transferred commands that are waiting for execution, that is commands stored in the CBuffers, will still be executed, and results generated by them will be stored in the appropriate RFIFO.<br><br><b>NOTE:</b> CFINVx must not be written unless the MODEx is configured to disabled, and CFIFO status is IDLE.                                                                                                                                                                                                                                                           |

Table continues on the next page...

**EQADC\_CFCR2 field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | 0 No effect.<br>1 Invalidate all of the entries in the corresponding CFIFO.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 23<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 24–27<br>MODE5    | <p>CFIFO Operation Mode 5</p> <p>The MODEx field selects the CFIFO operation mode for CFIFOx. See <a href="#">CFIFO scan trigger modes</a> for more information on CFIFO trigger mode.</p> <p><b>NOTE:</b> If MODEx is not disabled, it must not be changed to any other mode besides disabled. If MODEx is disabled and the CFIFO status is IDLE, MODEx can be changed to any other mode.</p> <p>0000 Disabled<br/> 0001 Software Trigger, Single Scan<br/> 0010 Low Level Gated External Trigger, Single Scan<br/> 0011 High Level Gated External Trigger, Single Scan<br/> 0100 Falling Edge External Trigger, Single Scan<br/> 0101 Rising Edge External Trigger, Single Scan<br/> 0110 Falling or Rising Edge External Trigger, Single Scan<br/> 0111 Reserved<br/> 1000 Reserved<br/> 1001 Software Trigger, Continuous Scan<br/> 1010 Low Level Gated External Trigger, Continuous Scan<br/> 1011 High Level Gated External Trigger, Continuous Scan<br/> 1100 Falling Edge External Trigger, Continuous Scan<br/> 1101 Rising Edge External Trigger, Continuous Scan<br/> 1110 Falling or Rising Edge External Trigger, Continuous Scan<br/> 1111 Reserved</p> |
| 28–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

## 29.5.9 EQADC Interrupt and DMA Control Register 0 (EQADC\_IDCR0)

The EQADC Interrupt Control Registers (EQADC\_IDCR) contain bits to enable the generation of interrupt or DMA requests when the corresponding flag bits are set in [EQADC FIFO and Interrupt Status Register \(EQADC\\_FISR<sub>n</sub>\)](#).

Address: 0h base + 60h offset = 60h

| Bit   | 0     | 1      | 2    | 3     | 4      | 5 | 6     | 7     | 8 | 9 | 10 | 11 | 12     | 13 | 14    | 15    |
|-------|-------|--------|------|-------|--------|---|-------|-------|---|---|----|----|--------|----|-------|-------|
| R     | NCIE0 | TORIE0 | PIE0 | EQIE0 | CFUIE0 | 0 | CFFE0 | CFFS0 | 0 |   |    |    | RFOIE0 | 0  | RFDE0 | RFDS0 |
| W     |       |        |      |       |        |   |       |       |   |   |    |    |        |    |       |       |
| Reset | 0     | 0      | 0    | 0     | 0      | 0 | 0     | 0     | 0 | 0 | 0  | 0  | 0      | 0  | 0     | 0     |

| Bit   | 16    | 17     | 18   | 19    | 20     | 21 | 22    | 23    | 24 | 25 | 26 | 27 | 28     | 29 | 30    | 31    |
|-------|-------|--------|------|-------|--------|----|-------|-------|----|----|----|----|--------|----|-------|-------|
| R     | NCIE1 | TORIE1 | PIE1 | EQIE1 | CFUIE1 | 0  | CFFE1 | CFFS1 | 0  |    |    |    | RFOIE1 | 0  | RFDE1 | RFDS1 |
| W     |       |        |      |       |        |    |       |       |    |    |    |    |        |    |       |       |
| Reset | 0     | 0      | 0    | 0     | 0      | 0  | 0     | 0     | 0  | 0  | 0  | 0  | 0      | 0  | 0     | 0     |

### EQADC\_IDCR0 field descriptions

| Field       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>NCIE0  | <p>Non-Coherency Interrupt Enable 0</p> <p>NCIE<sub>x</sub> enables the EQADC to generate an interrupt request when the corresponding NCF<sub>x</sub> in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISR<sub>n</sub>)</a> is asserted.</p> <p>0 Disable non-coherency interrupt request.<br/>1 Enable non-coherency interrupt request.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 1<br>TORIE0 | <p>Trigger Overrun Interrupt Enable 0</p> <p>TORIE<sub>x</sub> enables the EQADC to generate an interrupt request when the corresponding TORF<sub>x</sub> in <a href="#">EQADC DMA/interrupt request</a> is asserted.</p> <p>Apart from generating an independent interrupt request for a CFIFO<sub>x</sub> Trigger Overrun event, the EQADC also provides a combined interrupt at which the Result FIFO Overflow Interrupt, the Command FIFO Underflow Interrupt, and the Command FIFO Trigger Overrun Interrupt requests of ALL CFIFOs are ORed. When RFOIE<sub>x</sub>, CFUIE<sub>x</sub>, and TORIE<sub>x</sub> are all asserted, this combined interrupt request is asserted whenever one of the following 18 flags becomes asserted: RFOF<sub>x</sub>, CFUF<sub>x</sub>, and TORF<sub>x</sub> (assuming that all interrupts are enabled). See <a href="#">EQADC DMA/interrupt request</a> for details.</p> <p>0 Disable trigger overrun interrupt request.<br/>1 Enable trigger overrun interrupt request.</p> |
| 2<br>PIE0   | <p>Pause Interrupt Enable 0</p> <p>PIE<sub>x</sub> enables the EQADC to generate an interrupt request when the corresponding PF<sub>x</sub> in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISR<sub>n</sub>)</a> is asserted.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

Table continues on the next page...



**EQADC\_IDCR0 field descriptions (continued)**

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                  | 0 Disable pause interrupt request.<br>1 Enable pause interrupt request.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 3<br>EOQIE0      | End of Queue Interrupt Enable 0<br>EOQIEEx enables the EQADC to generate an interrupt request when the corresponding EOQFx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> is asserted.<br>0 Disable End of Queue interrupt request.<br>1 Enable End of Queue interrupt request.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 4<br>CFUIE0      | CFIFO Underflow Interrupt Enable 0<br>CFUIEEx enables the EQADC to generate an interrupt request when the corresponding CFUFx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> is asserted.<br>Apart from generating an independent interrupt request for a CFIFOx underflow event, the EQADC also provides a combined interrupt at which the Result FIFO Overflow Interrupt, the Command FIFO Underflow Interrupt, and the Command FIFO Trigger Overrun Interrupt requests of ALL CFIFOs are ORed. When RFOIEEx, CFUIEEx, and TORIEEx are all asserted, this combined interrupt request is asserted whenever one of the following 18 flags becomes asserted: RFOFx, CFUFx, and TORFx (assuming that all interrupts are enabled). See <a href="#">EQADC DMA/interrupt request</a> for details.<br>0 Disable Underflow Interrupt request.<br>1 Enable Underflow Interrupt request. |
| 5<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 6<br>CFFE0       | CFIFO Fill Enable 0<br>CFFEx enables the EQADC to generate an interrupt request (CFFSx is negated) or DMA request (CFFSx is asserted) when CFFFx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> is asserted.<br><b>NOTE:</b> CFFEx must not be negated while a DMA transaction is in progress.<br>0 Disable CFIFO Fill DMA or Interrupt request.<br>1 Enable CFIFO Fill DMA or Interrupt request.                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 7<br>CFFS0       | CFIFO Fill Select 0<br>CFFSx selects if a DMA or interrupt request is generated when CFFFx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> is asserted. If CFFEx is asserted, the EQADC generates an interrupt request when CFFSx is negated, or it generates a DMA request if CFFSx is asserted.<br><b>NOTE:</b> CFFSx must not be negated while a DMA transaction is in progress.<br>0 Generate interrupt request to move data from the system memory to CFIFOx.<br>1 Generate DMA request to move data from the system memory to CFIFOx.                                                                                                                                                                                                                                                                                                                                      |
| 8–11<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 12<br>RFOIE0     | RFIFO Overflow Interrupt Enable 0<br>RFOIEEx enables the EQADC to generate an interrupt request when the corresponding RFOFx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> is asserted.<br>Apart from generating an independent interrupt request for an RFIFOx overflow event, the EQADC also provides a combined interrupt at which the Result FIFO Overflow Interrupt, the Command FIFO Underflow Interrupt, and the Command FIFO Trigger Overrun Interrupt requests of ALL CFIFOs are ORed. When                                                                                                                                                                                                                                                                                                                                                                           |

*Table continues on the next page...*

## EQADC\_IDCR0 field descriptions (continued)

| Field          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | <p>RFOIEx, CFUIEx, and TORIEx are all asserted, this combined interrupt request is asserted whenever one of the following 18 flags becomes asserted: RFOFx, CFUFx, and TORFx (assuming that all interrupts are enabled). See <a href="#">EQADC DMA/interrupt request</a> for details.</p> <p>0 Disable Overflow Interrupt request.<br/>1 Enable Overflow Interrupt request.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 13<br>Reserved | <p>This field is reserved.<br/>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 14<br>RFDE0    | <p>RFIFO Drain Enable 0</p> <p>RFDEx enables the EQADC to generate an interrupt request (RFDSx is negated) or DMA request (RFDSx is asserted) when RFDFx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> is asserted.</p> <p><b>NOTE:</b> RFDEx must not be negated while a DMA transaction is in progress.</p> <p>0 Disable RFIFO Drain DMA or Interrupt request.<br/>1 Enable RFIFO Drain DMA or Interrupt request.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 15<br>RFDS0    | <p>RFIFO Drain Select 0</p> <p>RFDSx selects if a DMA or interrupt request is generated when RFDFx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> is asserted. If RFDEx is asserted, the EQADC generates an interrupt request when RFDSx is negated, or it generates a DMA request when RFDSx is asserted.</p> <p><b>NOTE:</b> RFDSx must not be negated while a DMA transaction is in progress.</p> <p>0 Generate interrupt request to move data from RFIFOx to the system memory.<br/>1 Generate DMA request to move data from RFIFOx to the system memory.</p>                                                                                                                                                                                                                                                                                                                                                 |
| 16<br>NCIE1    | <p>Non-Coherency Interrupt Enable 1</p> <p>NCIEx enables the EQADC to generate an interrupt request when the corresponding NCFx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> is asserted.</p> <p>0 Disable non-coherency interrupt request.<br/>1 Enable non-coherency interrupt request.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 17<br>TORIE1   | <p>Trigger Overrun Interrupt Enable 1</p> <p>TORIEx enables the EQADC to generate an interrupt request when the corresponding TORFx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> is asserted.</p> <p>Apart from generating an independent interrupt request for a CFIFOx Trigger Overrun event, the EQADC also provides a combined interrupt at which the Result FIFO Overflow Interrupt, the Command FIFO Underflow Interrupt, and the Command FIFO Trigger Overrun Interrupt requests of ALL CFIFOs are ORed. When RFOIEx, CFUIEx, and TORIEx are all asserted, this combined interrupt request is asserted whenever one of the following 18 flags becomes asserted: RFOFx, CFUFx, and TORFx (assuming that all interrupts are enabled). See <a href="#">EQADC DMA/interrupt request</a> for details.</p> <p>0 Disable trigger overrun interrupt request.<br/>1 Enable trigger overrun interrupt request.</p> |
| 18<br>PIE1     | <p>Pause Interrupt Enable 1</p> <p>PIEx enables the EQADC to generate an interrupt request when the corresponding PFx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> is asserted.</p> <p>0 Disable pause interrupt request.<br/>1 Enable pause interrupt request.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

Table continues on the next page...

**EQADC\_IDCR0 field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 19<br>EOQIE1      | <p>End of Queue Interrupt Enable 1</p> <p>EOQIE<sub>x</sub> enables the EQADC to generate an interrupt request when the corresponding EOQF<sub>x</sub> in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISR<sub>n</sub>)</a> is asserted.</p> <p>0 Disable End of Queue interrupt request.<br/>1 Enable End of Queue interrupt request.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 20<br>CFUIE1      | <p>CFIFO Underflow Interrupt Enable 1</p> <p>CFUIE<sub>x</sub> enables the EQADC to generate an interrupt request when the corresponding CFUF<sub>x</sub> in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISR<sub>n</sub>)</a> is asserted.</p> <p>Apart from generating an independent interrupt request for a CFIFO<sub>x</sub> underflow event, the EQADC also provides a combined interrupt at which the Result FIFO Overflow Interrupt, the Command FIFO Underflow Interrupt, and the Command FIFO Trigger Overrun Interrupt requests of ALL CFIFOs are ORed. When RFOIE<sub>x</sub>, CFUIE<sub>x</sub>, and TORIE<sub>x</sub> are all asserted, this combined interrupt request is asserted whenever one of the following 18 flags becomes asserted: RFOF<sub>x</sub>, CFUF<sub>x</sub>, and TORF<sub>x</sub> (assuming that all interrupts are enabled). See <a href="#">EQADC DMA/interrupt request</a> for details.</p> <p>0 Disable Underflow Interrupt request.<br/>1 Enable Underflow Interrupt request.</p> |
| 21<br>Reserved    | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 22<br>CFFE1       | <p>CFIFO Fill Enable 1</p> <p>CFFE<sub>x</sub> enables the EQADC to generate an interrupt request (CFFS<sub>x</sub> is negated) or DMA request (CFFS<sub>x</sub> is asserted) when CFFF<sub>x</sub> in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISR<sub>n</sub>)</a> is asserted.</p> <p><b>NOTE:</b> CFFE<sub>x</sub> must not be negated while a DMA transaction is in progress.</p> <p>0 Disable CFIFO Fill DMA or Interrupt request.<br/>1 Enable CFIFO Fill DMA or Interrupt request.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 23<br>CFFS1       | <p>CFIFO Fill Select 1</p> <p>CFFS<sub>x</sub> selects if a DMA or interrupt request is generated when CFFF<sub>x</sub> in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISR<sub>n</sub>)</a> is asserted. If CFFE<sub>x</sub> is asserted, the EQADC generates an interrupt request when CFFS<sub>x</sub> is negated, or it generates a DMA request if CFFS<sub>x</sub> is asserted.</p> <p><b>NOTE:</b> CFFS<sub>x</sub> must not be negated while a DMA transaction is in progress.</p> <p>0 Generate interrupt request to move data from the system memory to CFIFO<sub>x</sub>.<br/>1 Generate DMA request to move data from the system memory to CFIFO<sub>x</sub>.</p>                                                                                                                                                                                                                                                                                                                                             |
| 24–27<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 28<br>RFOIE1      | <p>RFIFO Overflow Interrupt Enable 1</p> <p>RFOIE<sub>x</sub> enables the EQADC to generate an interrupt request when the corresponding RFOF<sub>x</sub> in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISR<sub>n</sub>)</a> is asserted.</p> <p>Apart from generating an independent interrupt request for an RFIFO<sub>x</sub> overflow event, the EQADC also provides a combined interrupt at which the Result FIFO Overflow Interrupt, the Command FIFO Underflow Interrupt, and the Command FIFO Trigger Overrun Interrupt requests of ALL CFIFOs are ORed. When RFOIE<sub>x</sub>, CFUIE<sub>x</sub>, and TORIE<sub>x</sub> are all asserted, this combined interrupt request is asserted whenever one of the following 18 flags becomes asserted: RFOF<sub>x</sub>, CFUF<sub>x</sub>, and TORF<sub>x</sub> (assuming that all interrupts are enabled). See <a href="#">EQADC DMA/interrupt request</a> for details.</p>                                                                                          |

*Table continues on the next page...*

## EQADC\_IDCR0 field descriptions (continued)

| Field          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | 0 Disable Overflow Interrupt request.<br>1 Enable Overflow Interrupt request.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 29<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 30<br>RFDE1    | RFIFO Drain Enable 1<br>RFDEx enables the EQADC to generate an interrupt request (RFDSx is negated) or DMA request (RFDSx is asserted) when RFDFx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISR<sub>n</sub>)</a> is asserted.<br><b>NOTE:</b> RFDEx must not be negated while a DMA transaction is in progress.<br>0 Disable RFIFO Drain DMA or Interrupt request.<br>1 Enable RFIFO Drain DMA or Interrupt request.                                                                                                                                          |
| 31<br>RFDS1    | RFIFO Drain Select 1<br>RFDSx selects if a DMA or interrupt request is generated when RFDFx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISR<sub>n</sub>)</a> is asserted. If RFDEx is asserted, the EQADC generates an interrupt request when RFDSx is negated, or it generates a DMA request when RFDSx is asserted.<br><b>NOTE:</b> RFDSx must not be negated while a DMA transaction is in progress.<br>0 Generate interrupt request to move data from RFIFOx to the system memory.<br>1 Generate DMA request to move data from RFIFOx to the system memory. |

## 29.5.10 EQADC Interrupt and DMA Control Register 1 (EQADC\_IDCR1)

The EQADC Interrupt Control Registers (EQADC\_IDCR) contain bits to enable the generation of interrupt or DMA requests when the corresponding flag bits are set in [EQADC FIFO and Interrupt Status Register \(EQADC\\_FISR<sub>n</sub>\)](#).

Address: 0h base + 64h offset = 64h

|       |       |        |      |        |        |   |       |       |   |   |    |    |        |    |       |       |
|-------|-------|--------|------|--------|--------|---|-------|-------|---|---|----|----|--------|----|-------|-------|
| Bit   | 0     | 1      | 2    | 3      | 4      | 5 | 6     | 7     | 8 | 9 | 10 | 11 | 12     | 13 | 14    | 15    |
| R     | NCIE2 | TORIE2 | PIE2 | EOQIE2 | CFUIE2 | 0 | CFFE2 | CFFS2 | 0 |   |    |    | RFOIE2 | 0  | RFDE2 | RFDS2 |
| W     |       |        |      |        |        |   |       |       |   |   |    |    |        |    |       |       |
| Reset | 0     | 0      | 0    | 0      | 0      | 0 | 0     | 0     | 0 | 0 | 0  | 0  | 0      | 0  | 0     | 0     |

|       |       |        |      |        |        |    |       |       |    |    |    |    |        |    |       |       |
|-------|-------|--------|------|--------|--------|----|-------|-------|----|----|----|----|--------|----|-------|-------|
| Bit   | 16    | 17     | 18   | 19     | 20     | 21 | 22    | 23    | 24 | 25 | 26 | 27 | 28     | 29 | 30    | 31    |
| R     | NCIE3 | TORIE3 | PIE3 | EOQIE3 | CFUIE3 | 0  | CFFE3 | CFFS3 | 0  |    |    |    | RFOIE3 | 0  | RFDE3 | RFDS3 |
| W     |       |        |      |        |        |    |       |       |    |    |    |    |        |    |       |       |
| Reset | 0     | 0      | 0    | 0      | 0      | 0  | 0     | 0     | 0  | 0  | 0  | 0  | 0      | 0  | 0     | 0     |

## EQADC\_IDCR1 field descriptions

| Field         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>NCIE2    | <p>Non-Coherency Interrupt Enable 2</p> <p>NCIEEx enables the EQADC to generate an interrupt request when the corresponding NCFx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> is asserted.</p> <p>0 Disable non-coherency interrupt request.<br/>1 Enable non-coherency interrupt request.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 1<br>TORIE2   | <p>Trigger Overrun Interrupt Enable 2</p> <p>TORIEEx enables the EQADC to generate an interrupt request when the corresponding TORFx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> is asserted.</p> <p>Apart from generating an independent interrupt request for a CFIFOx Trigger Overrun event, the EQADC also provides a combined interrupt at which the Result FIFO Overflow Interrupt, the Command FIFO Underflow Interrupt, and the Command FIFO Trigger Overrun Interrupt requests of ALL CFIFOs are ORed. When RFOIEEx, CFUIEx, and TORIEEx are all asserted, this combined interrupt request is asserted whenever one of the following 18 flags becomes asserted: RFOFx, CFUFx, and TORFx (assuming that all interrupts are enabled). See <a href="#">EQADC DMA/interrupt request</a> for details.</p> <p>0 Disable trigger overrun interrupt request.<br/>1 Enable trigger overrun interrupt request.</p> |
| 2<br>PIE2     | <p>Pause Interrupt Enable 2</p> <p>PIEEx enables the EQADC to generate an interrupt request when the corresponding PFx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> is asserted.</p> <p>0 Disable pause interrupt request.<br/>1 Enable pause interrupt request.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 3<br>EOQIE2   | <p>End of Queue Interrupt Enable 2</p> <p>EOQIEEx enables the EQADC to generate an interrupt request when the corresponding EOQFx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> is asserted.</p> <p>0 Disable End of Queue interrupt request.<br/>1 Enable End of Queue interrupt request.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 4<br>CFUIE2   | <p>CFIFO Underflow Interrupt Enable 2</p> <p>CFUIEEx enables the EQADC to generate an interrupt request when the corresponding CFUFx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> is asserted.</p> <p>Apart from generating an independent interrupt request for a CFIFOx underflow event, the EQADC also provides a combined interrupt at which the Result FIFO Overflow Interrupt, the Command FIFO Underflow Interrupt, and the Command FIFO Trigger Overrun Interrupt requests of ALL CFIFOs are ORed. When RFOIEEx, CFUIEx, and TORIEEx are all asserted, this combined interrupt request is asserted whenever one of the following 18 flags becomes asserted: RFOFx, CFUFx, and TORFx (assuming that all interrupts are enabled). See <a href="#">EQADC DMA/interrupt request</a> for details.</p> <p>0 Disable Underflow Interrupt request.<br/>1 Enable Underflow Interrupt request.</p>                   |
| 5<br>Reserved | <p>This field is reserved.<br/>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 6<br>CFFE2    | <p>CFIFO Fill Enable 2</p> <p>CFFEx enables the EQADC to generate an interrupt request (CFFSx is negated) or DMA request (CFFSx is asserted) when CFFFx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> is asserted.</p> <p><b>NOTE:</b> CFFEx must not be negated while a DMA transaction is in progress.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |

*Table continues on the next page...*

## EQADC\_IDCR1 field descriptions (continued)

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                  | 0 Disable CFIFO Fill DMA or Interrupt request.<br>1 Enable CFIFO Fill DMA or Interrupt request.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 7<br>CFFS2       | CFIFO Fill Select 2<br><br>CFFSx selects if a DMA or interrupt request is generated when CFFx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> is asserted. If CFFEx is asserted, the EQADC generates an interrupt request when CFFSx is negated, or it generates a DMA request if CFFSx is asserted.<br><br><b>NOTE:</b> CFFSx must not be negated while a DMA transaction is in progress.<br><br>0 Generate interrupt request to move data from the system memory to CFIFOx.<br>1 Generate DMA request to move data from the system memory to CFIFOx.                                                                                                                                                                                                                                                                                                                                |
| 8–11<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 12<br>RFOIE2     | RFIFO Overflow Interrupt Enable 2<br><br>RFOIEx enables the EQADC to generate an interrupt request when the corresponding RFOFx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> is asserted.<br><br>Apart from generating an independent interrupt request for an RFIFOx overflow event, the EQADC also provides a combined interrupt at which the Result FIFO Overflow Interrupt, the Command FIFO Underflow Interrupt, and the Command FIFO Trigger Overrun Interrupt requests of ALL CFIFOs are ORed. When RFOIEx, CFUIEx, and TORIEx are all asserted, this combined interrupt request is asserted whenever one of the following 18 flags becomes asserted: RFOFx, CFUFx, and TORFx (assuming that all interrupts are enabled). See <a href="#">EQADC DMA/interrupt request</a> for details.<br><br>0 Disable Overflow Interrupt request.<br>1 Enable Overflow Interrupt request. |
| 13<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 14<br>RFDE2      | RFIFO Drain Enable 2<br><br>RFDEx enables the EQADC to generate an interrupt request (RFDSx is negated) or DMA request (RFDSx is asserted) when RFDFx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> is asserted.<br><br><b>NOTE:</b> RFDEx must not be negated while a DMA transaction is in progress.<br><br>0 Disable RFIFO Drain DMA or Interrupt request.<br>1 Enable RFIFO Drain DMA or Interrupt request.                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 15<br>RFDS2      | RFIFO Drain Select 2<br><br>RFDSx selects if a DMA or interrupt request is generated when RFDFx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> is asserted. If RFDEx is asserted, the EQADC generates an interrupt request when RFDSx is negated, or it generates a DMA request when RFDSx is asserted.<br><br><b>NOTE:</b> RFDSx must not be negated while a DMA transaction is in progress.<br><br>0 Generate interrupt request to move data from RFIFOx to the system memory.<br>1 Generate DMA request to move data from RFIFOx to the system memory.                                                                                                                                                                                                                                                                                                                            |
| 16<br>NCIE3      | Non-Coherency Interrupt Enable 3<br><br>NCIEx enables the EQADC to generate an interrupt request when the corresponding NCFx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> is asserted.<br><br>0 Disable non-coherency interrupt request.<br>1 Enable non-coherency interrupt request.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

Table continues on the next page...

**EQADC\_IDCR1 field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 17<br>TORIE3   | <p>Trigger Overrun Interrupt Enable 3</p> <p>TORIE<sub>x</sub> enables the EQADC to generate an interrupt request when the corresponding TORF<sub>x</sub> in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISR<sub>n</sub>)</a> is asserted.</p> <p>Apart from generating an independent interrupt request for a CFIFO<sub>x</sub> Trigger Overrun event, the EQADC also provides a combined interrupt at which the Result FIFO Overflow Interrupt, the Command FIFO Underflow Interrupt, and the Command FIFO Trigger Overrun Interrupt requests of ALL CFIFOs are ORed. When RFOIE<sub>x</sub>, CFUIE<sub>x</sub>, and TORIE<sub>x</sub> are all asserted, this combined interrupt request is asserted whenever one of the following 18 flags becomes asserted: RFOF<sub>x</sub>, CFUF<sub>x</sub>, and TORF<sub>x</sub> (assuming that all interrupts are enabled). See <a href="#">EQADC DMA/interrupt request</a> for details.</p> <p>0 Disable trigger overrun interrupt request.<br/>1 Enable trigger overrun interrupt request.</p> |
| 18<br>PIE3     | <p>Pause Interrupt Enable 3</p> <p>PIE<sub>x</sub> enables the EQADC to generate an interrupt request when the corresponding PF<sub>x</sub> in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISR<sub>n</sub>)</a> is asserted.</p> <p>0 Disable pause interrupt request.<br/>1 Enable pause interrupt request.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 19<br>EOQIE3   | <p>End of Queue Interrupt Enable 3</p> <p>EOQIE<sub>x</sub> enables the EQADC to generate an interrupt request when the corresponding EOQF<sub>x</sub> in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISR<sub>n</sub>)</a> is asserted.</p> <p>0 Disable End of Queue interrupt request.<br/>1 Enable End of Queue interrupt request.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 20<br>CFUIE3   | <p>CFIFO Underflow Interrupt Enable 3</p> <p>CFUIE<sub>x</sub> enables the EQADC to generate an interrupt request when the corresponding CFUF<sub>x</sub> in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISR<sub>n</sub>)</a> is asserted.</p> <p>Apart from generating an independent interrupt request for a CFIFO<sub>x</sub> underflow event, the EQADC also provides a combined interrupt at which the Result FIFO Overflow Interrupt, the Command FIFO Underflow Interrupt, and the Command FIFO Trigger Overrun Interrupt requests of ALL CFIFOs are ORed. When RFOIE<sub>x</sub>, CFUIE<sub>x</sub>, and TORIE<sub>x</sub> are all asserted, this combined interrupt request is asserted whenever one of the following 18 flags becomes asserted: RFOF<sub>x</sub>, CFUF<sub>x</sub>, and TORF<sub>x</sub> (assuming that all interrupts are enabled). See <a href="#">EQADC DMA/interrupt request</a> for details.</p> <p>0 Disable Underflow Interrupt request.<br/>1 Enable Underflow Interrupt request.</p>                   |
| 21<br>Reserved | <p>This field is reserved.<br/>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 22<br>CFFE3    | <p>CFIFO Fill Enable 3</p> <p>CFFE<sub>x</sub> enables the EQADC to generate an interrupt request (CFFS<sub>x</sub> is negated) or DMA request (CFFS<sub>x</sub> is asserted) when CFFF<sub>x</sub> in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISR<sub>n</sub>)</a> is asserted.</p> <p><b>NOTE:</b> CFFE<sub>x</sub> must not be negated while a DMA transaction is in progress.</p> <p>0 Disable CFIFO Fill DMA or Interrupt request.<br/>1 Enable CFIFO Fill DMA or Interrupt request.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 23<br>CFFS3    | <p>CFIFO Fill Select 3</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

*Table continues on the next page...*



## EQADC\_IDCR1 field descriptions (continued)

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | <p>CFFSx selects if a DMA or interrupt request is generated when CFFEx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> is asserted. If CFFEx is asserted, the EQADC generates an interrupt request when CFFSx is negated, or it generates a DMA request if CFFSx is asserted.</p> <p><b>NOTE:</b> CFFSx must not be negated while a DMA transaction is in progress.</p> <p>0 Generate interrupt request to move data from the system memory to CFIFOx.<br/>1 Generate DMA request to move data from the system memory to CFIFOx.</p>                                                                                                                                                                                                                                                                                                                                                          |
| 24–27<br>Reserved | <p>This field is reserved.<br/>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 28<br>RFOIE3      | <p>RFIFO Overflow Interrupt Enable 3</p> <p>RFOIEx enables the EQADC to generate an interrupt request when the corresponding RFOFx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> is asserted.</p> <p>Apart from generating an independent interrupt request for an RFIFOx overflow event, the EQADC also provides a combined interrupt at which the Result FIFO Overflow Interrupt, the Command FIFO Underflow Interrupt, and the Command FIFO Trigger Overrun Interrupt requests of ALL CFIFOs are ORed. When RFOIEx, CFUIEx, and TORIEx are all asserted, this combined interrupt request is asserted whenever one of the following 18 flags becomes asserted: RFOFx, CFUFx, and TORFx (assuming that all interrupts are enabled). See <a href="#">EQADC DMA/interrupt request</a> for details.</p> <p>0 Disable Overflow Interrupt request.<br/>1 Enable Overflow Interrupt request.</p> |
| 29<br>Reserved    | <p>This field is reserved.<br/>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 30<br>RFDE3       | <p>RFIFO Drain Enable 3</p> <p>RFDEx enables the EQADC to generate an interrupt request (RFDSx is negated) or DMA request (RFDSx is asserted) when RFDFx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> is asserted.</p> <p><b>NOTE:</b> RFDEx must not be negated while a DMA transaction is in progress.</p> <p>0 Disable RFIFO Drain DMA or Interrupt request.<br/>1 Enable RFIFO Drain DMA or Interrupt request.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 31<br>RFDS3       | <p>RFIFO Drain Select 3</p> <p>RFDSx selects if a DMA or interrupt request is generated when RFDFx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> is asserted. If RFDEx is asserted, the EQADC generates an interrupt request when RFDSx is negated, or it generates a DMA request when RFDSx is asserted.</p> <p><b>NOTE:</b> RFDSx must not be negated while a DMA transaction is in progress.</p> <p>0 Generate interrupt request to move data from RFIFOx to the system memory.<br/>1 Generate DMA request to move data from RFIFOx to the system memory.</p>                                                                                                                                                                                                                                                                                                                            |



### 29.5.11 EQADC Interrupt and DMA Control Register 2 (EQADC\_IDCR2)

The EQADC Interrupt Control Registers (EQADC\_IDCR) contain bits to enable the generation of interrupt or DMA requests when the corresponding flag bits are set in [EQADC FIFO and Interrupt Status Register \(EQADC\\_FISR<sub>n</sub>\)](#).

Address: 0h base + 68h offset = 68h

| Bit   | 0     | 1      | 2    | 3      | 4      | 5 | 6     | 7     | 8 | 9 | 10 | 11 | 12     | 13 | 14    | 15    |
|-------|-------|--------|------|--------|--------|---|-------|-------|---|---|----|----|--------|----|-------|-------|
| R     | NCIE4 | TORIE4 | PIE4 | EOQIE4 | CFUIE4 | 0 | CFIE4 | CFIS4 | 0 |   |    |    | RFOIE4 | 0  | RFDE4 | RFDS4 |
| W     |       |        |      |        |        |   |       |       |   |   |    |    |        |    |       |       |
| Reset | 0     | 0      | 0    | 0      | 0      | 0 | 0     | 0     | 0 | 0 | 0  | 0  | 0      | 0  | 0     | 0     |

| Bit   | 16    | 17     | 18   | 19     | 20     | 21 | 22    | 23    | 24 | 25 | 26 | 27 | 28     | 29 | 30    | 31    |
|-------|-------|--------|------|--------|--------|----|-------|-------|----|----|----|----|--------|----|-------|-------|
| R     | NCIE5 | TORIE5 | PIE5 | EOQIE5 | CFUIE5 | 0  | CFIE5 | CFIS5 | 0  |    |    |    | RFOIE5 | 0  | RFDE5 | RFDS5 |
| W     |       |        |      |        |        |    |       |       |    |    |    |    |        |    |       |       |
| Reset | 0     | 0      | 0    | 0      | 0      | 0  | 0     | 0     | 0  | 0  | 0  | 0  | 0      | 0  | 0     | 0     |

**EQADC\_IDCR2 field descriptions**

| Field       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>NCIE4  | <p>Non-Coherency Interrupt Enable 4</p> <p>NCIE<sub>x</sub> enables the EQADC to generate an interrupt request when the corresponding NCF<sub>x</sub> in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISR<sub>n</sub>)</a> is asserted.</p> <p>0 Disable non-coherency interrupt request.<br/>1 Enable non-coherency interrupt request.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 1<br>TORIE4 | <p>Trigger Overrun Interrupt Enable 4</p> <p>TORIE<sub>x</sub> enables the EQADC to generate an interrupt request when the corresponding TORF<sub>x</sub> in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISR<sub>n</sub>)</a> is asserted.</p> <p>Apart from generating an independent interrupt request for a CFIFOF<sub>x</sub> Trigger Overrun event, the EQADC also provides a combined interrupt at which the Result FIFO Overflow Interrupt, the Command FIFO Underflow Interrupt, and the Command FIFO Trigger Overrun Interrupt requests of ALL CFIFOs are ORed. When RFOIE<sub>x</sub>, CFUIE<sub>x</sub>, and TORIE<sub>x</sub> are all asserted, this combined interrupt request is asserted whenever one of the following 18 flags becomes asserted: RFOF<sub>x</sub>, CFUF<sub>x</sub>, and TORF<sub>x</sub> (assuming that all interrupts are enabled). See <a href="#">EQADC DMA/interrupt request</a> for details.</p> <p>0 Disable trigger overrun interrupt request.<br/>1 Enable trigger overrun interrupt request.</p> |
| 2<br>PIE4   | <p>Pause Interrupt Enable 4</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

Table continues on the next page...

## EQADC\_IDCR2 field descriptions (continued)

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                  | <p>PIEx enables the EQADC to generate an interrupt request when the corresponding PFx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> is asserted.</p> <p>0 Disable pause interrupt request.</p> <p>1 Enable pause interrupt request.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 3<br>EOQIE4      | <p>End of Queue Interrupt Enable 4</p> <p>EOQIEEx enables the EQADC to generate an interrupt request when the corresponding EOQF<sub>x</sub> in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> is asserted.</p> <p>0 Disable End of Queue interrupt request.</p> <p>1 Enable End of Queue interrupt request.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 4<br>CFUIE4      | <p>CFIFO Underflow Interrupt Enable 4</p> <p>CFUIEx enables the EQADC to generate an interrupt request when the corresponding CFUF<sub>x</sub> in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> is asserted.</p> <p>Apart from generating an independent interrupt request for a CFIFO<sub>x</sub> underflow event, the EQADC also provides a combined interrupt at which the Result FIFO Overflow Interrupt, the Command FIFO Underflow Interrupt, and the Command FIFO Trigger Overrun Interrupt requests of ALL CFIFOs are ORed. When RFOIE<sub>x</sub>, CFUIEx, and TORIE<sub>x</sub> are all asserted, this combined interrupt request is asserted whenever one of the following 18 flags becomes asserted: RFOF<sub>x</sub>, CFUF<sub>x</sub>, and TORF<sub>x</sub> (assuming that all interrupts are enabled). See <a href="#">EQADC DMA/interrupt request</a> for details.</p> <p>0 Disable Underflow Interrupt request.</p> <p>1 Enable Underflow Interrupt request.</p> |
| 5<br>Reserved    | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 6<br>CFFE4       | <p>CFIFO Fill Enable 4</p> <p>CFFEx enables the EQADC to generate an interrupt request (CFFS<sub>x</sub> is negated) or DMA request (CFFS<sub>x</sub> is asserted) when CFFF<sub>x</sub> in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> is asserted.</p> <p><b>NOTE:</b> CFFEx must not be negated while a DMA transaction is in progress.</p> <p>0 Disable CFIFO Fill DMA or Interrupt request.</p> <p>1 Enable CFIFO Fill DMA or Interrupt request.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 7<br>CFFS4       | <p>CFIFO Fill Select 4</p> <p>CFFS<sub>x</sub> selects if a DMA or interrupt request is generated when CFFF<sub>x</sub> in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> is asserted. If CFFEx is asserted, the EQADC generates an interrupt request when CFFS<sub>x</sub> is negated, or it generates a DMA request if CFFS<sub>x</sub> is asserted.</p> <p><b>NOTE:</b> CFFS<sub>x</sub> must not be negated while a DMA transaction is in progress.</p> <p>0 Generate interrupt request to move data from the system memory to CFIFO<sub>x</sub>.</p> <p>1 Generate DMA request to move data from the system memory to CFIFO<sub>x</sub>.</p>                                                                                                                                                                                                                                                                                                                                  |
| 8–11<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 12<br>RFOIE4     | <p>RFIFO Overflow Interrupt Enable 4</p> <p>RFOIEEx enables the EQADC to generate an interrupt request when the corresponding RFOF<sub>x</sub> in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> is asserted.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

Table continues on the next page...

## EQADC\_IDCR2 field descriptions (continued)

| Field          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | <p>Apart from generating an independent interrupt request for an RFIFOx overflow event, the EQADC also provides a combined interrupt at which the Result FIFO Overflow Interrupt, the Command FIFO Underflow Interrupt, and the Command FIFO Trigger Overrun Interrupt requests of ALL CFIFOs are ORed. When RFOIEx, CFUIEx, and TORIEx are all asserted, this combined interrupt request is asserted whenever one of the following 18 flags becomes asserted: RFOF<sub>x</sub>, CFUF<sub>x</sub>, and TORF<sub>x</sub> (assuming that all interrupts are enabled). See <a href="#">EQADC DMA/interrupt request</a> for details.</p> <p>0 Disable Overflow Interrupt request.<br/>1 Enable Overflow Interrupt request.</p>                                                                                                                                                                                                                                                                                     |
| 13<br>Reserved | <p>This field is reserved.<br/>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 14<br>RFDE4    | <p>RFIFO Drain Enable 4</p> <p>RFDE<sub>x</sub> enables the EQADC to generate an interrupt request (RFDS<sub>x</sub> is negated) or DMA request (RFDS<sub>x</sub> is asserted) when RFDF<sub>x</sub> in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISR<sub>n</sub>)</a> is asserted.</p> <p><b>NOTE:</b> RFDE<sub>x</sub> must not be negated while a DMA transaction is in progress.</p> <p>0 Disable RFIFO Drain DMA or Interrupt request.<br/>1 Enable RFIFO Drain DMA or Interrupt request.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 15<br>RFDS4    | <p>RFIFO Drain Select 4</p> <p>RFDS<sub>x</sub> selects if a DMA or interrupt request is generated when RFDF<sub>x</sub> in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISR<sub>n</sub>)</a> is asserted. If RFDE<sub>x</sub> is asserted, the EQADC generates an interrupt request when RFDS<sub>x</sub> is negated, or it generates a DMA request when RFDS<sub>x</sub> is asserted.</p> <p><b>NOTE:</b> RFDS<sub>x</sub> must not be negated while a DMA transaction is in progress.</p> <p>0 Generate interrupt request to move data from RFIFOx to the system memory.<br/>1 Generate DMA request to move data from RFIFOx to the system memory.</p>                                                                                                                                                                                                                                                                                                                                      |
| 16<br>NCIE5    | <p>Non-Coherency Interrupt Enable 5</p> <p>NCIE<sub>x</sub> enables the EQADC to generate an interrupt request when the corresponding NCF<sub>x</sub> in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISR<sub>n</sub>)</a> is asserted.</p> <p>0 Disable non-coherency interrupt request.<br/>1 Enable non-coherency interrupt request.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 17<br>TORIE5   | <p>Trigger Overrun Interrupt Enable 5</p> <p>TORIE<sub>x</sub> enables the EQADC to generate an interrupt request when the corresponding TORF<sub>x</sub> in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISR<sub>n</sub>)</a> is asserted.</p> <p>Apart from generating an independent interrupt request for a CFIFOx Trigger Overrun event, the EQADC also provides a combined interrupt at which the Result FIFO Overflow Interrupt, the Command FIFO Underflow Interrupt, and the Command FIFO Trigger Overrun Interrupt requests of ALL CFIFOs are ORed. When RFOIEx, CFUIEx, and TORIEx are all asserted, this combined interrupt request is asserted whenever one of the following 18 flags becomes asserted: RFOF<sub>x</sub>, CFUF<sub>x</sub>, and TORF<sub>x</sub> (assuming that all interrupts are enabled). See <a href="#">EQADC DMA/interrupt request</a> for details.</p> <p>0 Disable trigger overrun interrupt request.<br/>1 Enable trigger overrun interrupt request.</p> |
| 18<br>PIE5     | <p>Pause Interrupt Enable 5</p> <p>PIE<sub>x</sub> enables the EQADC to generate an interrupt request when the corresponding PF<sub>x</sub> in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISR<sub>n</sub>)</a> is asserted.</p> <p>0 Disable pause interrupt request.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

Table continues on the next page...

## EQADC\_IDCR2 field descriptions (continued)

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | 1 Enable pause interrupt request.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 19<br>EOQIE5      | <p>End of Queue Interrupt Enable 5</p> <p>EOQIE5 enables the EQADC to generate an interrupt request when the corresponding EOQFx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> is asserted.</p> <p>0 Disable End of Queue interrupt request.<br/>1 Enable End of Queue interrupt request.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 20<br>CFUIE5      | <p>CFIFO Underflow Interrupt Enable 5</p> <p>CFUIE5 enables the EQADC to generate an interrupt request when the corresponding CFUFx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> is asserted.</p> <p>Apart from generating an independent interrupt request for a CFIFOx underflow event, the EQADC also provides a combined interrupt at which the Result FIFO Overflow Interrupt, the Command FIFO Underflow Interrupt, and the Command FIFO Trigger Overrun Interrupt requests of ALL CFIFOs are ORed. When RFOIE5, CFUIE5, and TORIE5 are all asserted, this combined interrupt request is asserted whenever one of the following 18 flags becomes asserted: RFOFx, CFUFx, and TORFx (assuming that all interrupts are enabled). See <a href="#">EQADC DMA/interrupt request</a> for details.</p> <p>0 Disable Underflow Interrupt request.<br/>1 Enable Underflow Interrupt request.</p> |
| 21<br>Reserved    | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 22<br>CFFE5       | <p>CFIFO Fill Enable 5</p> <p>CFFEx enables the EQADC to generate an interrupt request (CFFSx is negated) or DMA request (CFFSx is asserted) when CFFFx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> is asserted.</p> <p><b>NOTE:</b> CFFEx must not be negated while a DMA transaction is in progress.</p> <p>0 Disable CFIFO Fill DMA or Interrupt request.<br/>1 Enable CFIFO Fill DMA or Interrupt request.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 23<br>CFFS5       | <p>CFIFO Fill Select 5</p> <p>CFFSx selects if a DMA or interrupt request is generated when CFFFx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> is asserted. If CFFEx is asserted, the EQADC generates an interrupt request when CFFSx is negated, or it generates a DMA request if CFFSx is asserted.</p> <p><b>NOTE:</b> CFFSx must not be negated while a DMA transaction is in progress.</p> <p>0 Generate interrupt request to move data from the system memory to CFIFOx.<br/>1 Generate DMA request to move data from the system memory to CFIFOx.</p>                                                                                                                                                                                                                                                                                                                                  |
| 24–27<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 28<br>RFOIE5      | <p>RFIFO Overflow Interrupt Enable 5</p> <p>RFOIE5 enables the EQADC to generate an interrupt request when the corresponding RFOFx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> is asserted.</p> <p>Apart from generating an independent interrupt request for an RFIFOx overflow event, the EQADC also provides a combined interrupt at which the Result FIFO Overflow Interrupt, the Command FIFO Underflow Interrupt, and the Command FIFO Trigger Overrun Interrupt requests of ALL CFIFOs are ORed. When RFOIE5, CFUIE5, and TORIE5 are all asserted, this combined interrupt request is asserted whenever one of the following 18 flags becomes asserted: RFOFx, CFUFx, and TORFx (assuming that all interrupts are enabled). See <a href="#">EQADC DMA/interrupt request</a> for details.</p>                                                                                          |

Table continues on the next page...

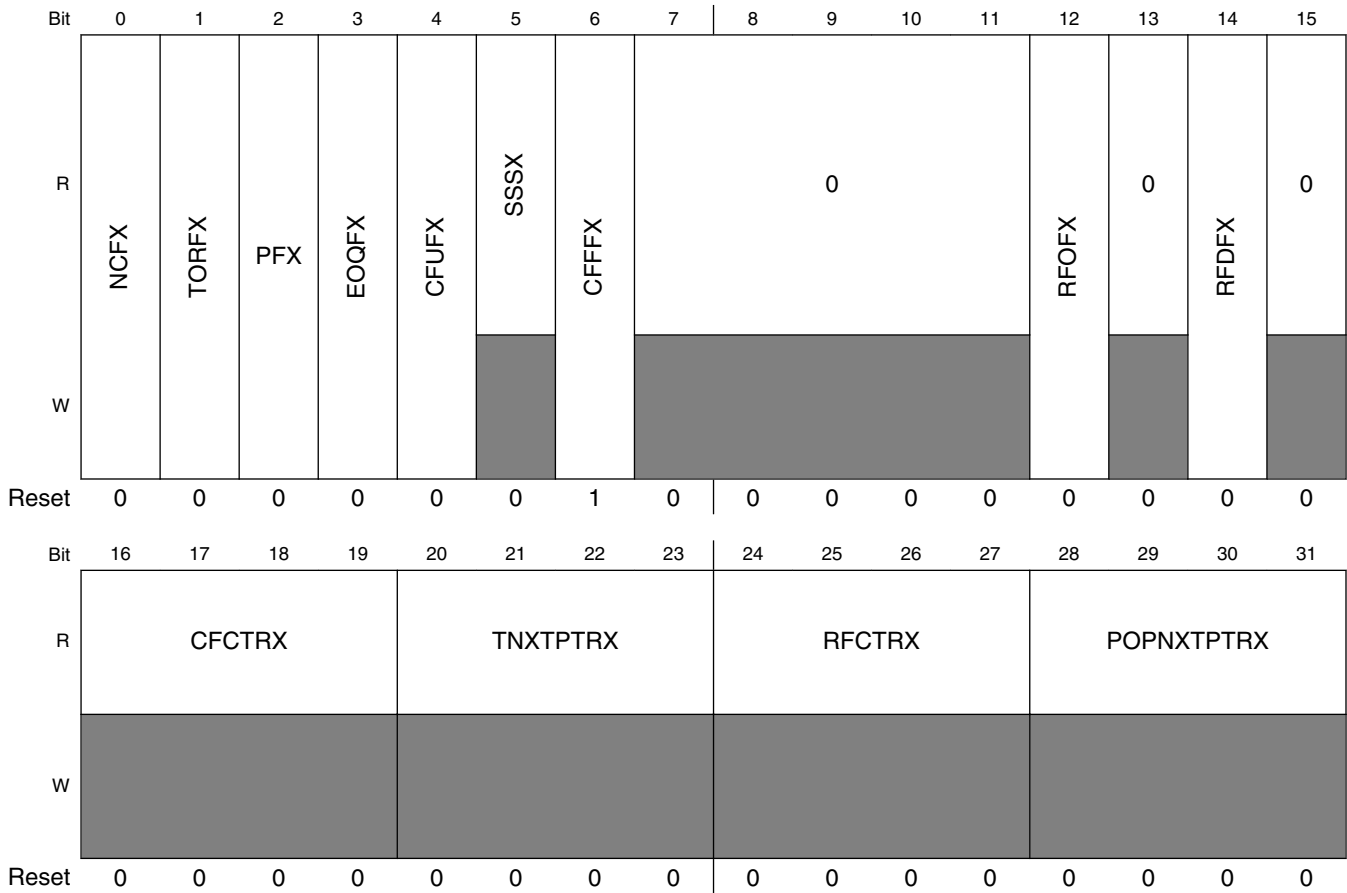
**EQADC\_IDCR2 field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | 0 Disable Overflow Interrupt request.<br>1 Enable Overflow Interrupt request.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 29<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 30<br>RFDE5    | RFIFO Drain Enable 5<br>RFDEx enables the EQADC to generate an interrupt request (RFDSx is negated) or DMA request (RFDSx is asserted) when RFDFx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> is asserted.<br><b>NOTE:</b> RFDEx must not be negated while a DMA transaction is in progress.<br>0 Disable RFIFO Drain DMA or Interrupt request.<br>1 Enable RFIFO Drain DMA or Interrupt request.                                                                                                                                          |
| 31<br>RFDS5    | RFIFO Drain Select 5<br>RFDSx selects if a DMA or interrupt request is generated when RFDFx in <a href="#">EQADC FIFO and Interrupt Status Register (EQADC_FISRn)</a> is asserted. If RFDEx is asserted, the EQADC generates an interrupt request when RFDSx is negated, or it generates a DMA request when RFDSx is asserted.<br><b>NOTE:</b> RFDSx must not be negated while a DMA transaction is in progress.<br>0 Generate interrupt request to move data from RFIFOx to the system memory.<br>1 Generate DMA request to move data from RFIFOx to the system memory. |

29.5.12 EQADC FIFO and Interrupt Status Register (EQADC\_FISRn)

The EQADC FIFO and Interrupt Status Registers (EQADC\_FISR) contain flag and status bits for each CFIFO and RFIFO pair. Write "1" to a flag bit to clear it. Writing "0" has no effect. Status bits are read only. These bits indicate the status of the FIFO itself.

Address: 0h base + 70h offset + (4d × i), where i=0d to 5d



EQADC\_FISRn field descriptions

| Field     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>NCFX | <div>Non-Coherency Flag</div> <div>NCFx is set whenever a command sequence being transferred through CFIFOx becomes non coherent. If NCIE<sub>x</sub> in <a href="#">EQADC Interrupt and DMA Control Register 0 (EQADC_IDCR0)</a> and NCF<sub>x</sub> are asserted, an interrupt request will be generated. Write "1" to clear NCF<sub>x</sub>. Writing a "0" has no effect. For more information refer to <a href="#">Command sequence non-coherency detection</a> .</div> <div>0 Command sequence being transferred by CFIFO<sub>x</sub> is coherent.</div> <div>1 Command sequence being transferred by CFIFO<sub>x</sub> became non-coherent.</div> |

Table continues on the next page...

EQADC\_FISR<sub>n</sub> field descriptions (continued)

| Field      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1<br>TORFX | <p>Trigger Overrun Flag for CFIFO<sub>x</sub></p> <p>TORFX is set when trigger overrun occurs for the specified CFIFO in edge or level trigger mode. Trigger overrun occurs when an already triggered CFIFO receives an additional trigger. When TORIE<sub>x</sub> in <a href="#">EQADC Interrupt and DMA Control Register 0 (EQADC_IDCR0)</a> and TORFX are asserted, an interrupt request will be generated.</p> <p>Apart from generating an independent interrupt request for a CFIFO<sub>x</sub> Trigger Overrun event, the EQADC also provides a combined interrupt at which the Result FIFO Overflow Interrupt, the Command FIFO Underflow Interrupt, and the Command FIFO Trigger Overrun Interrupt requests of ALL CFIFOs are ORed. When RFOIE<sub>x</sub>, CFUIE<sub>x</sub>, and TORIE<sub>x</sub> are all asserted, this combined interrupt request is asserted whenever one of the following 18 flags becomes asserted: RFOFX, CFUFx, and TORFX (assuming that all interrupts are enabled). See <a href="#">EQADC DMA/interrupt request</a> for details.</p> <p>Write "1" to clear the TORFX bit. Writing a "0" has no effect.</p> <p><b>NOTE:</b> The trigger overrun flag will not set for CFIFOs configured for software trigger mode.</p> <p>0 No trigger overrun occurred.<br/>1 Trigger overrun occurred.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 2<br>PFX   | <p>Pause Flag x</p> <p>PF behavior changes according to the CFIFO trigger mode. In edge trigger mode, PFX is set when the EQADC completes the transfer of an entry with an asserted Pause bit from CFIFO<sub>x</sub>. In level trigger mode, when CFIFO<sub>x</sub> is in TRIGGERED status, PFX is set when CFIFO status changes from TRIGGERED due to the detection of a closed gate. An interrupt routine, generated due to the asserted PF, can be used to verify if a complete scan of the CQueue was performed. If a closed gate is detected while no command transfers are taking place, it will have immediate effect on the CFIFO status. If a closed gate is detected while a command transfer to an on-chip CBuffer is taking place, it will only affect the CFIFO status when the transfer completes. The transfer of entries bound for the on-chip ADCs is considered completed when they are stored in the appropriate CBuffer. In software trigger mode, PFX will never become asserted.</p> <p>If PIE<sub>x</sub> in <a href="#">EQADC Interrupt and DMA Control Register 0 (EQADC_IDCR0)</a> and PFX are asserted, an interrupt will be generated. Write "1" to clear the PFX. Writing a "0" has no effect. See <a href="#">Pause status</a> for more information on the Pause Flag.</p> <p><b>NOTE:</b> In edge trigger mode, an asserted PFX only implies that the EQADC has finished transferring a command with an asserted PAUSE bit from CFIFO<sub>x</sub>. It does not imply that result data for the current command and for all previously transferred commands has been returned to the appropriate RFIFO.</p> <p><b>NOTE:</b> In software or level trigger mode, when the EQADC completes the transfer of an entry from CFIFO<sub>x</sub> with an asserted PAUSE bit, PFX will not be set and transfer of commands will continue without pausing.</p> <p>0 Entry with asserted PAUSE bit was not transferred from CFIFO<sub>x</sub> (CFIFO in edge trigger mode), or CFIFO status did not change from TRIGGERED due to detection of a closed gate (CFIFO in level trigger mode).<br/>1 Entry with asserted PAUSE bit was transferred from CFIFO<sub>x</sub> (CFIFO in edge trigger mode), or CFIFO status changes from TRIGGERED due to detection of a closed gate (CFIFO in level trigger mode).</p> |
| 3<br>EOQFX | <p>End of Queue Flag x</p> <p>EOQFX indicates that an entry with an asserted EOQ bit was transferred from CFIFO<sub>x</sub> to the on-chip ADCs - see <a href="#">Message format in EQADC</a> for details about command message formats. When the EQADC completes the transfer of an entry with an asserted EOQ bit from CFIFO<sub>x</sub>, EOQFX will be set. The transfer of entries bound for the on-chip ADCs is considered completed when they are stored in the appropriate</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

Table continues on the next page...

EQADC\_FISR<sub>n</sub> field descriptions (continued)

| Field                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                        | <p>CBuffer. If the EOQIE<sub>x</sub> bit in <a href="#">EQADC Interrupt and DMA Control Register 0 (EQADC_IDCR0)</a> and EOQF<sub>x</sub> are asserted, an interrupt will be generated. Write "1" to clear the EOQF<sub>x</sub> bit. Writing a "0" has no effect. Refer to <a href="#">CQueue completion status</a> for more information on the End of Queue Flag.</p> <p><b>NOTE:</b> An asserted EOQF<sub>x</sub> only implies that the EQADC has finished transferring a command with an asserted EOQ bit from CFIFO<sub>x</sub>. It does not imply that result data for the current command and for all previously transferred commands has been returned to the appropriate RFIFO.</p> <p>0 Entry with asserted EOQ bit was not transferred from CFIFO<sub>x</sub>.<br/>1 Entry with asserted EOQ bit was transferred from CFIFO<sub>x</sub>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 4<br>CFUF <sub>x</sub> | <p>CFIFO Underflow Flag x</p> <p>CFUF<sub>x</sub> indicates an underflow event on CFIFO<sub>x</sub>. CFUF<sub>x</sub> is set when CFIFO<sub>x</sub> is in TRIGGERED state and it becomes empty. No commands will be transferred from an underflowing CFIFO, nor will command transfers from lower priority CFIFOs be blocked. When CFUIE<sub>x</sub> in <a href="#">EQADC Interrupt and DMA Control Register 0 (EQADC_IDCR0)</a> and CFUF<sub>x</sub> are both asserted, the EQADC generates an interrupt request.</p> <p>Apart from generating an independent interrupt request for a CFIFO<sub>x</sub> underflow event, the EQADC also provides a combined interrupt at which the Result FIFO Overflow Interrupt, the Command FIFO Underflow Interrupt, and the Command FIFO Trigger Overrun Interrupt requests of ALL CFIFOs are ORed. When RFOIE<sub>x</sub>, CFUIE<sub>x</sub>, and TORIE<sub>x</sub> are all asserted, this combined interrupt request is asserted whenever one of the following 18 flags becomes asserted: RFOF<sub>x</sub>, CFUF<sub>x</sub>, and TORF<sub>x</sub> (assuming that all interrupts are enabled). See <a href="#">EQADC DMA/interrupt request</a> for details.</p> <p>Write "1" to clear CFUF<sub>x</sub>. Writing a "0" has no effect.</p> <p>0 No CFIFO underflow event occurred.<br/>1 A CFIFO underflow event occurred.</p> |
| 5<br>SSS <sub>x</sub>  | <p>CFIFO Single-Scan Status Bit x</p> <p>An asserted SSS<sub>x</sub> bit enables the detection of trigger events for CFIFOs programmed into single-scan level- or edge-trigger mode, and works as trigger for CFIFOs programmed into single-scan software-trigger mode. See <a href="#">Single-scan mode</a> for further details. The SSS<sub>x</sub> bit is set by writing a "1" to the SSE<sub>x</sub> bit in <a href="#">EQADC CFIFO Control Register 0 (EQADC_CFCR0)</a>. The EQADC clears the SSS<sub>x</sub> bit when a command with an asserted EOQ bit is transferred from a CFIFO in single-scan mode, when a CFIFO is in single-scan level-trigger mode and its status changes from TRIGGERED due to the detection of a closed gate, or when the value of the CFIFO operation mode (MODE<sub>x</sub>) in <a href="#">EQADC CFIFO Control Register 0 (EQADC_CFCR0)</a> is changed to disabled. Writing to SSS<sub>x</sub> has no effect. SSS<sub>x</sub> has no effect in continuous-scan or in disabled mode.</p> <p>0 CFIFO in single-scan level- or edge-trigger mode will ignore trigger events, or CFIFO in single-scan software-trigger mode is not triggered.<br/>1 CFIFO in single-scan level- or edge-trigger mode will detect a trigger event, or CFIFO in single-scan software-trigger mode is triggered.</p>                                    |
| 6<br>CFFF <sub>x</sub> | <p>CFIFO Fill Flag x</p> <p>CFFF<sub>x</sub> is set when the CFIFO<sub>x</sub> is not full. When CFFE<sub>x</sub> in <a href="#">EQADC Interrupt and DMA Control Register 0 (EQADC_IDCR0)</a> and CFFF<sub>x</sub> are both asserted, an interrupt or a DMA request will be generated depending on the status of the CFFS<sub>x</sub> bit. When CFFS<sub>x</sub> is negated (interrupt requests selected), software clears CFFF<sub>x</sub> by writing a "1" to it. Writing a "0" has no effect. When CFFS<sub>x</sub> is asserted (DMA requests selected), CFFF<sub>x</sub> is automatically cleared by the EQADC when the CFIFO becomes full.</p> <p><b>NOTE:</b> Writing "1" to CFFF<sub>x</sub> when CFFS<sub>x</sub> is asserted (DMA requests selected) is not allowed.</p> <p><b>NOTE:</b> When generation of interrupt requests is selected (CFFS<sub>x</sub>=0), CFFF<sub>x</sub> must only be cleared in the ISR after the CFIFO<sub>x</sub> push register is accessed.</p>                                                                                                                                                                                                                                                                                                                                                                                |

*Table continues on the next page...*



**EQADC\_FISR<sub>n</sub> field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | 0 CFIFO <sub>x</sub> is full.<br>1 CFIFO <sub>x</sub> is not full.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 7–11<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 12<br>RFOFX       | RFIFO Overflow Flag x<br><p>RFOFX indicates an overflow event on RFIFO<sub>x</sub>. RFOFX is set when RFIFO<sub>x</sub> is already full, and a new data is received from the on-chip ADCs. The RFIFO<sub>x</sub> will not overwrite older data in the RFIFO, and the new data will be ignored. When RFOIEx in <a href="#">EQADC Interrupt and DMA Control Register 0 (EQADC_IDCR0)</a> and RFOFX are both asserted, the EQADC generates an interrupt request.</p> <p>Apart from generating an independent interrupt request for an RFIFO<sub>x</sub> overflow event, the EQADC also provides a combined interrupt at which the Result FIFO Overflow Interrupt, the Command FIFO Underflow Interrupt, and the Command FIFO Trigger Overrun Interrupt requests of ALL CFIFOs are ORed. When RFOIEx, CFUIEx, and TORIEx are all asserted, this combined interrupt request is asserted whenever one of the following 18 flags becomes asserted: RFOFX, CFUFx, and TORFx (assuming that all interrupts are enabled). See <a href="#">EQADC DMA/interrupt request</a> for details.</p> <p>Write "1" to clear RFOFX. Writing a "0" has no effect.</p> 0 No RFIFO overflow event occurred.<br>1 An RFIFO overflow event occurred. |
| 13<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 14<br>RFDFX       | RFIFO Drain Flag x<br><p>RFDFx indicates if RFIFO<sub>x</sub> has valid entries or not. RFDFx is set when the RFIFO<sub>x</sub> has at least one valid entry in it. When RFDEx in <a href="#">EQADC Interrupt and DMA Control Register 0 (EQADC_IDCR0)</a> and RFDFx are both asserted, an interrupt or a DMA request will be generated depending on the status of the RFDSx bit. When RFDSx is negated (interrupt requests selected), software clears RFDFx by writing a "1" to it. Writing a "0" has no effect. When RFDSx is asserted (DMA requests selected), RFDFx is automatically cleared by the EQADC when the RFIFO becomes empty.</p> <p><b>NOTE:</b> Writing "1" to RFDFx when RFDSx is asserted (DMA requests selected) is not allowed.</p> <p><b>NOTE:</b> When the generation of interrupt requests is selected (RFDSx=0), RFDFx must only be cleared in the ISR after the RFIFOx pop register is accessed.</p> 0 RFIFO <sub>x</sub> is empty.<br>1 RFIFO <sub>x</sub> has at least one valid entry.                                                                                                                                                                                                        |
| 15<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 16–19<br>CFCTRX   | CFIFOx Entry Counter<br><p>CFCTRx indicates the number of commands stored in the CFIFO<sub>x</sub>. When the EQADC completes transferring a piece of new data from the CFIFO<sub>x</sub>, it decrements CFCTRx by one. Writing a word or any bytes to the corresponding <a href="#">EQADC CFIFO Push Register (EQADC_CFPR<sub>n</sub>)</a> increments CFCTRx by one. Writing any value to CFCTRx has no effect.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 20–23<br>TNXTPTRX | CFIFOx Transfer Next Pointer<br><p>TNXTPTRx indicates the index of the next entry to be removed from CFIFO<sub>x</sub> when it completes a transfer. When TNXTPTRx is zero, it points to the entry with the smallest memory-mapped address inside CFIFO<sub>x</sub>. TNXTPTRx is only updated when a command transfer is completed. If the maximum index number (CFIFO depth minus one) is reached, TNXTPTRx is wrapped to zero, else, it is incremented by one. For details, see <a href="#">CFIFO basic functionality</a>. Writing any value to TNXTPTRx has no effect.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

*Table continues on the next page...*

EQADC\_FISR<sub>n</sub> field descriptions (continued)

| Field                           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 24–27<br>RFCTR <sub>x</sub>     | <p>RFIFO<sub>x</sub> Entry Counter</p> <p>RFCTR<sub>x</sub> indicates the number of data items stored in the RFIFO<sub>x</sub>. When the EQADC stores a piece of new data into RFIFO<sub>x</sub>, it increments RFCTR<sub>x</sub> by one. Reading the whole word, half-word or any bytes of the corresponding <a href="#">EQADC Result FIFO Pop Register (EQADC_RFPR<sub>n</sub>)</a> decrements RFCTR<sub>x</sub> by one. Writing any value to RFCTR<sub>x</sub> itself has no effect.</p> <p><b>NOTE:</b> When the DMA request is selected ((RFDS<sub>x</sub>=1), the decrement of RFCTR<sub>x</sub> should be done only readings controlled by the DMA external block. Flushing of the RFIFO<sub>x</sub> decrements the RFCTR<sub>x</sub> to zero and can lead the DMA system to a locked state (DMA request ON).</p> |
| 28–31<br>POPNXTPTR <sub>x</sub> | <p>RFIFO<sub>x</sub> Pop Next Pointer</p> <p>POPNXTPTR<sub>x</sub> indicates the index of the entry that will be returned when EQADC_RFPR<sub>x</sub> is read. When POPNXTPTR<sub>x</sub> is zero, it points to the entry with the smallest memory-mapped address inside RFIFO<sub>x</sub>. POPNXTPTR<sub>x</sub> is updated when EQADC_RFPR<sub>x</sub> is read. If the maximum index number (RFIFO depth minus one) is reached, POPNXTPTR<sub>x</sub> is wrapped to zero, else, it is incremented by one. For details refer to <a href="#">RFIFO basic functionality</a>. Writing any value to POPNXTPTR<sub>x</sub> has no effect.</p>                                                                                                                                                                                |

### 29.5.13 EQADC CFIFO Transfer Counter Register 0 (EQADC\_CFTCR0)

The EQADC CFIFO Transfer Counter Registers (EQADC\_CFTCR) record the number of commands transferred from a CFIFO. The EQADC\_CFTCR supports the monitoring of command transfers from a CFIFO.

Address: 0h base + 90h offset = 90h

|       |   |   |   |   |   |        |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |
|-------|---|---|---|---|---|--------|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|--------|----|----|----|----|----|----|----|----|----|----|--|--|--|--|--|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5      | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21     | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |  |  |  |  |  |
| R     | 0 |   |   |   |   | TC_CF0 |   |   |   |   |    |    |    |    |    |    | 0  |    |    |    |    | TC_CF1 |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |
| W     |   |   |   |   |   |        |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |
| Reset | 0 | 0 | 0 | 0 | 0 | 0      | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0      | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |  |  |  |  |  |

## EQADC\_CFTCR0 field descriptions

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–4<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 5–15<br>TC_CF0  | <p>Transfer Counter for CFIFO0</p> <p>TC_CF<sub>x</sub> counts the number of commands which have been completely transferred from CFIFO<sub>x</sub>. The transfer of entries bound for the on-chip ADCs is considered completed when they are stored in the appropriate CBuffer. The EQADC increments the TC_CF<sub>x</sub> value by one after a command is transferred. TC_CF<sub>x</sub> resets to zero after EQADC completes transferring a command with an asserted EOQ bit. Writing any value to TC_CF<sub>x</sub> sets the counter to that written value.</p> <p><b>NOTE:</b> If CFIFO<sub>x</sub> is in TRIGGERED state when its MODE<sub>x</sub> field is programmed to disabled, the exact number of entries transferred from the CFIFO until that point - TC_CF<sub>x</sub> - is only known after the CFIFO status changes to IDLE, as indicated by CFS<sub>x</sub>. For details refer to <a href="#">Disabled Mode</a>.</p> |

Table continues on the next page...

**EQADC\_CFTCR0 field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 16–20<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 21–31<br>TC_CF1   | Transfer Counter for CFIFO1<br><br>TC_CFx counts the number of commands which have been completely transferred from CFIFOx. The transfer of entries bound for the on-chip ADCs is considered completed when they are stored in the appropriate CBuffer. The EQADC increments the TC_CFx value by one after a command is transferred. TC_CFx resets to zero after EQADC completes transferring a command with an asserted EOQ bit. Writing any value to TC_CFx sets the counter to that written value.<br><br><b>NOTE:</b> If CFIFOx is in TRIGGERED state when its MODEx field is programmed to disabled, the exact number of entries transferred from the CFIFO until that point - TC_CFx - is only known after the CFIFO status changes to IDLE, as indicated by CFSx. For details refer to <a href="#">Disabled Mode</a> . |

**29.5.14 EQADC CFIFO Transfer Counter Register 1 (EQADC\_CFTCR1)**

The EQADC CFIFO Transfer Counter Registers (EQADC\_CFTCR) record the number of commands transferred from a CFIFO. The EQADC\_CFTCR supports the monitoring of command transfers from a CFIFO.

Address: 0h base + 94h offset = 94h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**EQADC\_CFTCR1 field descriptions**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–4<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 5–15<br>TC_CF2    | Transfer Counter for CFIFO2<br><br>TC_CFx counts the number of commands which have been completely transferred from CFIFOx. The transfer of entries bound for the on-chip ADCs is considered completed when they are stored in the appropriate CBuffer. The EQADC increments the TC_CFx value by one after a command is transferred. TC_CFx resets to zero after EQADC completes transferring a command with an asserted EOQ bit. Writing any value to TC_CFx sets the counter to that written value.<br><br><b>NOTE:</b> If CFIFOx is in TRIGGERED state when its MODEx field is programmed to disabled, the exact number of entries transferred from the CFIFO until that point - TC_CFx - is only known after the CFIFO status changes to IDLE, as indicated by CFSx. For details refer to <a href="#">Disabled Mode</a> . |
| 16–20<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |

*Table continues on the next page...*

## EQADC\_CFTCR1 field descriptions (continued)

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 21–31<br>TC_CF3 | <p>Transfer Counter for CFIFO3</p> <p>TC_CFx counts the number of commands which have been completely transferred from CFIFOx. The transfer of entries bound for the on-chip ADCs is considered completed when they are stored in the appropriate CBuffer. The EQADC increments the TC_CFx value by one after a command is transferred. TC_CFx resets to zero after EQADC completes transferring a command with an asserted EOQ bit. Writing any value to TC_CFx sets the counter to that written value.</p> <p><b>NOTE:</b> If CFIFOx is in TRIGGERED state when its MODEx field is programmed to disabled, the exact number of entries transferred from the CFIFO until that point - TC_CFx - is only known after the CFIFO status changes to IDLE, as indicated by CFSx. For details refer to <a href="#">Disabled Mode</a>.</p> |

## 29.5.15 EQADC CFIFO Transfer Counter Register 2 (EQADC\_CFTCR2)

The EQADC CFIFO Transfer Counter Registers (EQADC\_CFTCR) record the number of commands transferred from a CFIFO. The EQADC\_CFTCR supports the monitoring of command transfers from a CFIFO.

Address: 0h base + 98h offset = 98h

|       |   |   |   |   |   |        |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |    |    |    |    |  |
|-------|---|---|---|---|---|--------|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|--------|----|----|----|----|----|----|----|----|----|--|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5      | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22     | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |  |
| R     | 0 |   |   |   |   | TC_CF4 |   |   |   |   |    |    |    |    |    |    |    | 0  |    |    |    |    | TC_CF5 |    |    |    |    |    |    |    |    |    |  |
| W     |   |   |   |   |   |        |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |    |    |    |    |  |
| Reset | 0 | 0 | 0 | 0 | 0 | 0      | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0      | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |  |

## EQADC\_CFTCR2 field descriptions

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–4<br>Reserved   | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 5–15<br>TC_CF4    | <p>Transfer Counter for CFIFO4</p> <p>TC_CFx counts the number of commands which have been completely transferred from CFIFOx. The transfer of entries bound for the on-chip ADCs is considered completed when they are stored in the appropriate CBuffer. The EQADC increments the TC_CFx value by one after a command is transferred. TC_CFx resets to zero after EQADC completes transferring a command with an asserted EOQ bit. Writing any value to TC_CFx sets the counter to that written value.</p> <p><b>NOTE:</b> If CFIFOx is in TRIGGERED state when its MODEx field is programmed to disabled, the exact number of entries transferred from the CFIFO until that point - TC_CFx - is only known after the CFIFO status changes to IDLE, as indicated by CFSx. For details refer to <a href="#">Disabled Mode</a>.</p> |
| 16–20<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 21–31<br>TC_CF5   | <p>Transfer Counter for CFIFO5</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

Table continues on the next page...

**EQADC\_CFTCR2 field descriptions (continued)**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | <p>TC_CFx counts the number of commands which have been completely transferred from CFIFOx. The transfer of entries bound for the on-chip ADCs is considered completed when they are stored in the appropriate CBuffer. The EQADC increments the TC_CFx value by one after a command is transferred. TC_CFx resets to zero after EQADC completes transferring a command with an asserted EOQ bit. Writing any value to TC_CFx sets the counter to that written value.</p> <p><b>NOTE:</b> If CFIFOx is in TRIGGERED state when its MODEx field is programmed to disabled, the exact number of entries transferred from the CFIFO until that point - TC_CFx - is only known after the CFIFO status changes to IDLE, as indicated by CFSx. For details refer to <a href="#">Disabled Mode</a>.</p> |

**29.5.16 EQADC CFIFO Status Snapshot Register 0 (EQADC\_CFSSR0)**

The EQADC CFIFO Status Snapshot Registers (EQADC\_CFSSR) contain status fields to track the operation status of each CFIFO and the transfer counter of the last CFIFO to initiate a command transfer to the internal CBuffers. EQADC\_CFSSR0-1 are related to the on-chip CBuffers (CBuffer0-1). All fields of a particular EQADC\_CFSSR register are captured at the beginning of a command transfer to the CBuffer associated with that register. Note that captured status register values are associated with previous command transfer. This means that the CFSSR registers capture the status registers before the status registers change because of the transfer of the current command that is about to be popped from the CFIFO. The EQADC\_CFSSR registers are read only. Writing to the EQADC\_CFSSR registers has no effect.

Address: 0h base + A0h offset = A0h

|       |           |           |           |           |           |            |    |    |    |    |    |    |    |    |    |    |
|-------|-----------|-----------|-----------|-----------|-----------|------------|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0         | 1         | 2         | 3         | 4         | 5          | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | CFS0_TCB0 | CFS1_TCB0 | CFS2_TCB0 | CFS3_TCB0 | CFS4_TCB0 | CFS5_TCB0  | 0  |    |    |    |    |    |    |    |    |    |
| W     |           |           |           |           |           |            |    |    |    |    |    |    |    |    |    |    |
| Reset | 0         | 0         | 0         | 0         | 0         | 0          | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| Bit   | 16        | 17        | 18        | 19        | 20        | 21         | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0         | LCFTCB0   |           |           |           | TC_LCFTCB0 |    |    |    |    |    |    |    |    |    |    |
| W     |           |           |           |           |           |            |    |    |    |    |    |    |    |    |    |    |
| Reset | 0         | 1         | 1         | 1         | 1         | 0          | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**EQADC\_CFSSR0 field descriptions**

| Field            | Description                                                                                                                                                                                                                                                                                                             |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–1<br>CFS0_TCB0 | <p>CFIFO Status at Transfer to CBuffer0</p> <p>CFSx_TCBn indicates the CFIFOx status of previously completed command transfer. CFSx_TCBn is a copy of the corresponding CFSx in <a href="#">EQADC CFIFO Status Register (EQADC_CFSR)</a>, captured at the time a current command transfer to CBuffern is initiated.</p> |

Table continues on the next page...

**EQADC\_CFSSR0 field descriptions (continued)**

| Field               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2–3<br>CFS1_TCB0    | CFIFO Status at Transfer to CBuffer0<br>CFSx_TCBn indicates the CFIFOx status of previously completed command transfer. CFSx_TCBn is a copy of the corresponding CFSx in <a href="#">EQADC CFIFO Status Register (EQADC_CFSR)</a> , captured at the time a current command transfer to CBuffern is initiated.                                                                                                                                                                                                                                                                                                                                     |
| 4–5<br>CFS2_TCB0    | CFIFO Status at Transfer to CBuffer0<br>CFSx_TCBn indicates the CFIFOx status of previously completed command transfer. CFSx_TCBn is a copy of the corresponding CFSx in <a href="#">EQADC CFIFO Status Register (EQADC_CFSR)</a> , captured at the time a current command transfer to CBuffern is initiated.                                                                                                                                                                                                                                                                                                                                     |
| 6–7<br>CFS3_TCB0    | CFIFO Status at Transfer to CBuffer0<br>CFSx_TCBn indicates the CFIFOx status of previously completed command transfer. CFSx_TCBn is a copy of the corresponding CFSx in <a href="#">EQADC CFIFO Status Register (EQADC_CFSR)</a> , captured at the time a current command transfer to CBuffern is initiated.                                                                                                                                                                                                                                                                                                                                     |
| 8–9<br>CFS4_TCB0    | CFIFO Status at Transfer to CBuffer0<br>CFSx_TCBn indicates the CFIFOx status of previously completed command transfer. CFSx_TCBn is a copy of the corresponding CFSx in <a href="#">EQADC CFIFO Status Register (EQADC_CFSR)</a> , captured at the time a current command transfer to CBuffern is initiated.                                                                                                                                                                                                                                                                                                                                     |
| 10–11<br>CFS5_TCB0  | CFIFO Status at Transfer to CBuffer0<br>CFSx_TCBn indicates the CFIFOx status of previously completed command transfer. CFSx_TCBn is a copy of the corresponding CFSx in <a href="#">EQADC CFIFO Status Register (EQADC_CFSR)</a> , captured at the time a current command transfer to CBuffern is initiated.                                                                                                                                                                                                                                                                                                                                     |
| 12–16<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 17–20<br>LCFTCB0    | Last CFIFO to Transfer to CBuffer0<br>LCFTCBn holds the CFIFO number to have completed a previous command transfer to CBuffern.<br><br>0000 Last command was transferred from CFIFO0<br>0001 Last command was transferred from CFIFO1<br>0010 Last command was transferred from CFIFO2<br>0011 Last command was transferred from CFIFO3<br>0100 Last command was transferred from CFIFO4<br>0101 Last command was transferred from CFIFO5<br>0110 Reserved<br>0111 Reserved<br>1000 Reserved<br>1001 Reserved<br>1010 Reserved<br>1011 Reserved<br>1100 Reserved<br>1101 Reserved<br>1110 Reserved<br>1111 No command was transferred to CBuffern |
| 21–31<br>TC_LCFTCB0 | Transfer Counter for Last CFIFO to transfer commands to CBuffer0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

*Table continues on the next page...*

**EQADC\_CFSSR0 field descriptions (continued)**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | TC_LCFTCBn indicates the number of commands which have been completely transferred from CFIFOx when a current command transfer from CFIFOx to CBuffern is initiated. TC_LCFTCBn is a copy of the corresponding TC_CFx in <a href="#">EQADC CFIFO Transfer Counter Register 0 (EQADC_CFTCR0)</a> , captured at the time a current command transfer from CFIFOx to CBuffern is initiated. This field has no meaning when LCFTCBn is 0b1111. |

**29.5.17 EQADC CFIFO Status Snapshot Register 1 (EQADC\_CFSSR1)**

The EQADC CFIFO Status Snapshot Registers (EQADC\_CFSSR) contain status fields to track the operation status of each CFIFO and the transfer counter of the last CFIFO to initiate a command transfer to the internal CBuffers. EQADC\_CFSSR0-1 are related to the on-chip CBuffers (CBuffer0-1). All fields of a particular EQADC\_CFSSR register are captured at the beginning of a command transfer to the CBuffer associated with that register. Note that captured status register values are associated with previous command transfer. This means that the CFSSR registers capture the status registers before the status registers change because of the transfer of the current command that is about to be popped from the CFIFO. The EQADC\_CFSSR registers are read only. Writing to the EQADC\_CFSSR registers has no effect.

Address: 0h base + A4h offset = A4h

|       |           |           |           |           |           |            |    |    |    |    |    |    |    |    |    |    |
|-------|-----------|-----------|-----------|-----------|-----------|------------|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0         | 1         | 2         | 3         | 4         | 5          | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | CFS0_TCB1 | CFS1_TCB1 | CFS2_TCB1 | CFS3_TCB1 | CFS4_TCB1 | CFS5_TCB1  | 0  |    |    |    |    |    |    |    |    |    |
| W     |           |           |           |           |           |            |    |    |    |    |    |    |    |    |    |    |
| Reset | 0         | 0         | 0         | 0         | 0         | 0          | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| Bit   | 16        | 17        | 18        | 19        | 20        | 21         | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0         | LCFTCB1   |           |           |           | TC_LCFTCB1 |    |    |    |    |    |    |    |    |    |    |
| W     |           |           |           |           |           |            |    |    |    |    |    |    |    |    |    |    |
| Reset | 0         | 1         | 1         | 1         | 1         | 0          | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**EQADC\_CFSSR1 field descriptions**

| Field            | Description                                                                                                                                                                                                                                                                                                       |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–1<br>CFS0_TCB1 | CFIFO Status at Transfer to CBuffer1<br><br>CFSx_TCBn indicates the CFIFOx status of previously completed command transfer. CFSx_TCBn is a copy of the corresponding CFSx in <a href="#">EQADC CFIFO Status Register (EQADC_CFSR)</a> , captured at the time a current command transfer to CBuffern is initiated. |
| 2–3<br>CFS1_TCB1 | CFIFO Status at Transfer to CBuffer1                                                                                                                                                                                                                                                                              |

*Table continues on the next page...*



**EQADC\_CFSSR1 field descriptions (continued)**

| Field               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                     | CFSx_TCBn indicates the CFIFOx status of previously completed command transfer. CFSx_TCBn is a copy of the corresponding CFSx in <a href="#">EQADC CFIFO Status Register (EQADC_CFSR)</a> , captured at the time a current command transfer to CBuffern is initiated.                                                                                                                                                                                                                                                                                                                                                                             |
| 4–5<br>CFS2_TCB1    | CFIFO Status at Transfer to CBuffer1<br>CFSx_TCBn indicates the CFIFOx status of previously completed command transfer. CFSx_TCBn is a copy of the corresponding CFSx in <a href="#">EQADC CFIFO Status Register (EQADC_CFSR)</a> , captured at the time a current command transfer to CBuffern is initiated.                                                                                                                                                                                                                                                                                                                                     |
| 6–7<br>CFS3_TCB1    | CFIFO Status at Transfer to CBuffer1<br>CFSx_TCBn indicates the CFIFOx status of previously completed command transfer. CFSx_TCBn is a copy of the corresponding CFSx in <a href="#">EQADC CFIFO Status Register (EQADC_CFSR)</a> , captured at the time a current command transfer to CBuffern is initiated.                                                                                                                                                                                                                                                                                                                                     |
| 8–9<br>CFS4_TCB1    | CFIFO Status at Transfer to CBuffer1<br>CFSx_TCBn indicates the CFIFOx status of previously completed command transfer. CFSx_TCBn is a copy of the corresponding CFSx in <a href="#">EQADC CFIFO Status Register (EQADC_CFSR)</a> , captured at the time a current command transfer to CBuffern is initiated.                                                                                                                                                                                                                                                                                                                                     |
| 10–11<br>CFS5_TCB1  | CFIFO Status at Transfer to CBuffer1<br>CFSx_TCBn indicates the CFIFOx status of previously completed command transfer. CFSx_TCBn is a copy of the corresponding CFSx in <a href="#">EQADC CFIFO Status Register (EQADC_CFSR)</a> , captured at the time a current command transfer to CBuffern is initiated.                                                                                                                                                                                                                                                                                                                                     |
| 12–16<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 17–20<br>LCFTCB1    | Last CFIFO to Transfer to CBuffer1<br>LCFTCBn holds the CFIFO number to have completed a previous command transfer to CBuffern.<br><br>0000 Last command was transferred from CFIFO0<br>0001 Last command was transferred from CFIFO1<br>0010 Last command was transferred from CFIFO2<br>0011 Last command was transferred from CFIFO3<br>0100 Last command was transferred from CFIFO4<br>0101 Last command was transferred from CFIFO5<br>0110 Reserved<br>0111 Reserved<br>1000 Reserved<br>1001 Reserved<br>1010 Reserved<br>1011 Reserved<br>1100 Reserved<br>1101 Reserved<br>1110 Reserved<br>1111 No command was transferred to CBuffern |
| 21–31<br>TC_LCFTCB1 | Transfer Counter for Last CFIFO to transfer commands to CBuffer1<br>TC_LCFTCBn indicates the number of commands which have been completely transferred from CFIFOx when a current command transfer from CFIFOx to CBuffern is initiated. TC_LCFTCBn is a copy of the corresponding TC_CFx in <a href="#">EQADC CFIFO Transfer Counter Register 0 (EQADC_CFTCR0)</a> , captured at the time a current command transfer from CFIFOx to CBuffern is initiated. This field has no meaning when LCFTCBn is 0b1111.                                                                                                                                     |



### 29.5.18 EQADC CFIFO Status Register (EQADC\_CFSR)

The EQADC CFIFO Status Register (EQADC\_CFSR) contains the current CFIFO status. The EQADC\_CFSR registers is read only. Writing to the EQADC\_CFSR register has no effect.

**Table 29-42. Current CFIFO Status**

| CFIFO Status        | Field Value | Explanation                                                                                                                                                                                                                                                                                                                                                  |
|---------------------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| IDLE                | 0b00        | CFIFO is disabled.<br>CFIFO is in single-scan edge or level trigger mode and does not have SSS asserted.<br>EQADC completed the transfer of the last entry of the CQueue in single-scan mode.                                                                                                                                                                |
| Reserved            | 0b01        | Not applicable.                                                                                                                                                                                                                                                                                                                                              |
| WAITING FOR TRIGGER | 0b10        | CFIFO Mode is modified to continuous-scan edge or level trigger mode.<br>CFIFO Mode is modified to single-scan edge or level trigger mode and SSS is asserted.<br>CFIFO Mode is modified to single-scan software trigger mode and SSS is negated.<br>CFIFO is paused.<br>EQADC transferred the last entry of the queue in continuous-scan edge trigger mode. |
| TRIGGERED           | 0b11        | CFIFO is triggered                                                                                                                                                                                                                                                                                                                                           |

Address: 0h base + ACh offset = ACh

|       |      |   |      |   |      |   |      |   |      |   |      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|------|---|------|---|------|---|------|---|------|---|------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0    | 1 | 2    | 3 | 4    | 5 | 6    | 7 | 8    | 9 | 10   | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | CFS0 |   | CFS1 |   | CFS2 |   | CFS3 |   | CFS4 |   | CFS5 |    | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |      |   |      |   |      |   |      |   |      |   |      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0    | 0 | 0    | 0 | 0    | 0 | 0    | 0 | 0    | 0 | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

#### EQADC\_CFSR field descriptions

| Field       | Description                                                                                                                        |
|-------------|------------------------------------------------------------------------------------------------------------------------------------|
| 0–1<br>CFS0 | CFIFO Status<br>CFSx indicates the current status of CFIFOx. See <a href="#">Table 29-42</a> for more information on CFIFO status. |
| 2–3<br>CFS1 | CFIFO Status<br>CFSx indicates the current status of CFIFOx. See <a href="#">Table 29-42</a> for more information on CFIFO status. |
| 4–5<br>CFS2 | CFIFO Status<br>CFSx indicates the current status of CFIFOx. See <a href="#">Table 29-42</a> for more information on CFIFO status. |
| 6–7<br>CFS3 | CFIFO Status<br>CFSx indicates the current status of CFIFOx. See <a href="#">Table 29-42</a> for more information on CFIFO status. |

*Table continues on the next page...*

**EQADC\_CFSR field descriptions (continued)**

| Field             | Description                                                                                                                        |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------|
| 8–9<br>CFS4       | CFIFO Status<br>CFSx indicates the current status of CFIFOx. See <a href="#">Table 29-42</a> for more information on CFIFO status. |
| 10–11<br>CFS5     | CFIFO Status<br>CFSx indicates the current status of CFIFOx. See <a href="#">Table 29-42</a> for more information on CFIFO status. |
| 12–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                            |

**29.5.19 EQADC STAC Bus Client Configuration Register (EQADC\_STACCCR)**

The EQADC STAC Bus Client Configuration Register (EQADC\_STACCCR) contains bits used to control which time slots the EQADC selects to obtain pre-defined external time bases.

**Table 29-44. STAC Bus timebase bits selection**

| STACBSm[ 0:3] | Selected Bits |
|---------------|---------------|
| 0b0000        | TBASEm[ 0:15] |
| 0b0001        | TBASEm[ 1:16] |
| 0b0010        | TBASEm[ 2:17] |
| 0b0011        | TBASEm[ 3:18] |
| 0b0100        | TBASEm[ 4:19] |
| 0b0101        | TBASEm[ 5:20] |
| 0b0110        | TBASEm[ 6:21] |
| 0b0111        | TBASEm[ 7:22] |
| 0b1000        | TBASEm[ 8:23] |
| Others        | Reserved      |

Address: 0h base + D0h offset = D0h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |         |    |    |    |      |    |    |    |         |    |    |    |      |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|---------|----|----|----|------|----|----|----|---------|----|----|----|------|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16      | 17 | 18 | 19 | 20   | 21 | 22 | 23 | 24      | 25 | 26 | 27 | 28   | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | STACBS2 |    |    |    | SRV2 |    |    |    | STACBS1 |    |    |    | SRV1 |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |         |    |    |    |      |    |    |    |         |    |    |    |      |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0       | 0  | 0  | 0  | 0    | 0  | 0  | 0  | 0       | 0  | 0  | 0  | 0    | 0  | 0  | 0  |

**EQADC\_STACCCR field descriptions**

| Field            | Description                                                                             |
|------------------|-----------------------------------------------------------------------------------------|
| 0–15<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |

*Table continues on the next page...*

**EQADC\_STACCCR field descriptions (continued)**

| Field            | Description                                                                                                                                                                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 16–19<br>STACBS2 | STAC Bus Timebase Bits Selection 2<br>The STACBSm field selects 16 bits from the total of 24 bits that are received from the STAC Bus interface as described in <a href="#">Table 29-44</a> . Consider TBASEm[ 0:23] the selected time base from slot SRVm. |
| 20–23<br>SRV2    | STAC Bus Server Data Slot Selector 2<br>The field SRVm indicates the slot number that contains the desired time base value sent by the STAC Bus server. It is possible to have up to 16 different sources to be selected.                                   |
| 24–27<br>STACBS1 | STAC Bus Timebase Bits Selection 1<br>The STACBSm field selects 16 bits from the total of 24 bits that are received from the STAC Bus interface as described in <a href="#">Table 29-44</a> . Consider TBASEm[ 0:23] the selected time base from slot SRVm. |
| 28–31<br>SRV1    | STAC Bus Server Data Slot Selector 1<br>The field SRVm indicates the slot number that contains the desired time base value sent by the STAC Bus server. It is possible to have up to 16 different sources to be selected.                                   |

**29.5.20 EQADC CFIFO0 Register (EQADC\_CF0Rn)**

The EQADC CFIFO Registers (EQADC\_CFxRw) (x=0, .., 5; w=0, .., 3) provide visibility of the contents of a CFIFO for debugging purposes. Each CFIFO has four registers which are uniquely mapped to its four 32-bit entries. See [EQADC command FIFOs](#) for more information on CFIFOs. These registers are read only. Data written to these registers is ignored.

Address: 0h base + 100h offset + (4d × i), where i=0d to 3d

|       |              |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|--------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0            | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | CFIFO0_DATAW |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |              |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0            | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**EQADC\_CF0Rn field descriptions**

| Field                | Description                                                                                                                                                                                                                               |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>CFIFO0_DATAW | CFIFO0 Data w (w = 0, .., 3)<br>Reading CFIFOx_DATAw returns the value stored on the wth entry of CFIFOx. Each CFIFO is composed of four 32-bit entries, with register 0 being mapped to the one with the smallest memory mapped address. |

## 29.5.21 EQADC CFIFO0 Extension Register (EQADC\_CF0ER<sub>n</sub>)

The EQADC CFIFO0 Extension Registers (EQADC\_CF0ER<sub>w</sub>) (w=0, ..., 3) provide visibility of the contents of the extended portion of CFIFO0 for debugging purposes. There are four registers which are uniquely mapped to its four 32-bit entries. See [EQADC command FIFOs](#) for more information on CFIFOs. These registers are read only. Data written to these registers is ignored.

Address: 0h base + 110h offset + (4d × i), where i=0d to 3d

|       |               |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0             | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | CFIFO0_EDATAW |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |               |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0             | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

### EQADC\_CF0ER<sub>n</sub> field descriptions

| Field                     | Description                                                                                                                                                                                                                                    |
|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>CFIFO0_<br>EDATAW | CFIFO0 Data w (w = 0, ..., 3)<br><br>Reading CFIFO0_DATAw returns the value stored on the wth entry of CFIFO0. Each CFIFO is composed of four 32-bit entries, with register 0 being mapped to the one with the smallest memory mapped address. |

## 29.5.22 EQADC CFIFO1 Register (EQADC\_CF1R<sub>n</sub>)

The EQADC CFIFO Registers (EQADC\_CF<sub>x</sub>R<sub>w</sub>) (x=0, ..., 5; w=0, ..., 3) provide visibility of the contents of a CFIFO for debugging purposes. Each CFIFO has four registers which are uniquely mapped to its four 32-bit entries. See [EQADC command FIFOs](#) for more information on CFIFOs. These registers are read only. Data written to these registers is ignored.

Address: 0h base + 140h offset + (4d × i), where i=0d to 3d

|       |              |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|--------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0            | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | CFIFO1_DATAW |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |              |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0            | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**EQADC\_CF1Rn field descriptions**

| Field                | Description                                                                                                                                                                                                                                |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>CFIFO1_DATAW | CFIFO1 Data w (w = 0, ..., 3)<br>Reading CFIFOx_DATAw returns the value stored on the wth entry of CFIFOx. Each CFIFO is composed of four 32-bit entries, with register 0 being mapped to the one with the smallest memory mapped address. |

**29.5.23 EQADC CFIFO2 Register (EQADC\_CF2Rn)**

The EQADC CFIFO Registers (EQADC\_CFxRw) (x=0, ..., 5; w=0, ..., 3) provide visibility of the contents of a CFIFO for debugging purposes. Each CFIFO has four registers which are uniquely mapped to its four 32-bit entries. See [EQADC command FIFOs](#) for more information on CFIFOs. These registers are read only. Data written to these registers is ignored.

Address: 0h base + 180h offset + (4d × i), where i=0d to 3d

|       |              |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|--------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0            | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | CFIFO2_DATAW |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |              |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0            | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

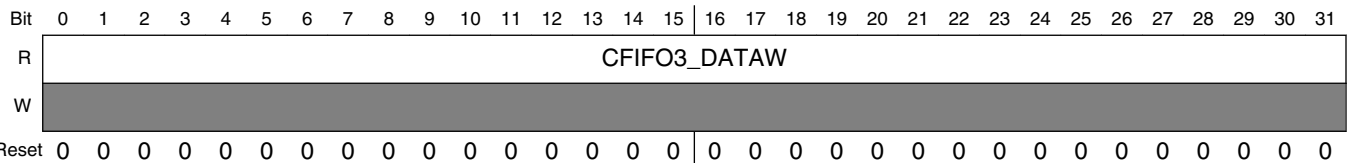
**EQADC\_CF2Rn field descriptions**

| Field                | Description                                                                                                                                                                                                                                |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>CFIFO2_DATAW | CFIFO2 Data w (w = 0, ..., 3)<br>Reading CFIFOx_DATAw returns the value stored on the wth entry of CFIFOx. Each CFIFO is composed of four 32-bit entries, with register 0 being mapped to the one with the smallest memory mapped address. |

29.5.24 EQADC CFIFO3 Register (EQADC\_CF3Rn)

The EQADC CFIFO Registers (EQADC\_CFxRw) (x=0, .., 5; w=0, .., 3) provide visibility of the contents of a CFIFO for debugging purposes. Each CFIFO has four registers which are uniquely mapped to its four 32-bit entries. See [EQADC command FIFOs](#) for more information on CFIFOs. These registers are read only. Data written to these registers is ignored.

Address: 0h base + 1C0h offset + (4d × i), where i=0d to 3d



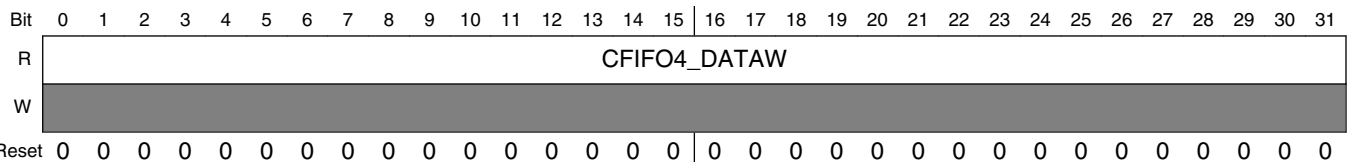
EQADC\_CF3Rn field descriptions

| Field                | Description                                                                                                                                                                                                                                    |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>CFIFO3_DATAW | CFIFO3 Data w (w = 0, ..., 3)<br><br>Reading CFIFOx_DATAw returns the value stored on the wth entry of CFIFOx. Each CFIFO is composed of four 32-bit entries, with register 0 being mapped to the one with the smallest memory mapped address. |

29.5.25 EQADC CFIFO4 Register (EQADC\_CF4Rn)

The EQADC CFIFO Registers (EQADC\_CFxRw) (x=0, .., 5; w=0, .., 3) provide visibility of the contents of a CFIFO for debugging purposes. Each CFIFO has four registers which are uniquely mapped to its four 32-bit entries. See [EQADC command FIFOs](#) for more information on CFIFOs. These registers are read only. Data written to these registers is ignored.

Address: 0h base + 200h offset + (4d × i), where i=0d to 3d



**EQADC\_CF4Rn field descriptions**

| Field                | Description                                                                                                                                                                                                                                |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>CFIFO4_DATAW | CFIFO4 Data w (w = 0, ..., 3)<br>Reading CFIFOx_DATAw returns the value stored on the wth entry of CFIFOx. Each CFIFO is composed of four 32-bit entries, with register 0 being mapped to the one with the smallest memory mapped address. |

**29.5.26 EQADC CFIFO5 Register (EQADC\_CF5Rn)**

The EQADC CFIFO Registers (EQADC\_CFxRw) (x=0, ..., 5; w=0, ..., 3) provide visibility of the contents of a CFIFO for debugging purposes. Each CFIFO has four registers which are uniquely mapped to its four 32-bit entries. See [EQADC command FIFOs](#) for more information on CFIFOs. These registers are read only. Data written to these registers is ignored.

Address: 0h base + 240h offset + (4d × i), where i=0d to 3d

|       |              |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|--------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0            | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | CFIFO5_DATAW |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |              |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0            | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**EQADC\_CF5Rn field descriptions**

| Field                | Description                                                                                                                                                                                                                                |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>CFIFO5_DATAW | CFIFO5 Data w (w = 0, ..., 3)<br>Reading CFIFOx_DATAw returns the value stored on the wth entry of CFIFOx. Each CFIFO is composed of four 32-bit entries, with register 0 being mapped to the one with the smallest memory mapped address. |

29.5.27 EQADC RFIFO0 Register (EQADC\_RF0Rn)

The EQADC RFIFO Registers (EQADC\_RFxRw) (x=0, .., 5; w=0, .., 3) provide visibility of the contents of a RFIFO for debugging purposes. Each RFIFO has four registers which are uniquely mapped to its four 16-bit entries. See [EQADC result FIFOs](#) for more information on RFIFOs. These registers are read only. Data written to these registers is ignored.

Address: 0h base + 300h offset + (4d × i), where i=0d to 3d

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16           | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | RFIFO0_DATAW |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0            | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

EQADC\_RF0Rn field descriptions

| Field                 | Description                                                                                                                                                                                                                               |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>Reserved      | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                   |
| 16–31<br>RFIFO0_DATAW | RFIFO0 Data w (w = 0, .., 3)<br>Reading RFIFOx_DATAw returns the value stored on the wth entry of RFIFOx. Each RFIFO is composed of four 16-bit entries, with register 0 being mapped to the one with the smallest memory mapped address. |

29.5.28 EQADC RFIFO1 Register (EQADC\_RF1Rn)

The EQADC RFIFO Registers (EQADC\_RFxRw) (x=0, .., 5; w=0, .., 3) provide visibility of the contents of a RFIFO for debugging purposes. Each RFIFO has four registers which are uniquely mapped to its four 16-bit entries. See [EQADC result FIFOs](#) for more information on RFIFOs. These registers are read only. Data written to these registers is ignored.

Address: 0h base + 340h offset + (4d × i), where i=0d to 3d

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16           | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | RFIFO1_DATAW |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0            | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |



**EQADC\_RF1Rn field descriptions**

| Field                 | Description                                                                                                                                                                                                                                |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>Reserved      | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                    |
| 16–31<br>RFIFO1_DATAW | RFIFO1 Data w (w = 0, ..., 3)<br>Reading RFIFOx_DATAw returns the value stored on the wth entry of RFIFOx. Each RFIFO is composed of four 16-bit entries, with register 0 being mapped to the one with the smallest memory mapped address. |

**29.5.29 EQADC RFIFO2 Register (EQADC\_RF2Rn)**

The EQADC RFIFO Registers (EQADC\_RFxRw) (x=0, ..., 5; w=0, ..., 3) provide visibility of the contents of a RFIFO for debugging purposes. Each RFIFO has four registers which are uniquely mapped to its four 16-bit entries. See [EQADC result FIFOs](#) for more information on RFIFOs. These registers are read only. Data written to these registers is ignored.

Address: 0h base + 380h offset + (4d × i), where i=0d to 3d

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16           | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | RFIFO2_DATAW |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0            | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**EQADC\_RF2Rn field descriptions**

| Field                 | Description                                                                                                                                                                                                                                |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>Reserved      | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                    |
| 16–31<br>RFIFO2_DATAW | RFIFO2 Data w (w = 0, ..., 3)<br>Reading RFIFOx_DATAw returns the value stored on the wth entry of RFIFOx. Each RFIFO is composed of four 16-bit entries, with register 0 being mapped to the one with the smallest memory mapped address. |

### 29.5.30 EQADC RFIFO3 Register (EQADC\_RF3Rn)

The EQADC RFIFO Registers (EQADC\_RFxRw) (x=0, ..., 5; w=0, ..., 3) provide visibility of the contents of a RFIFO for debugging purposes. Each RFIFO has four registers which are uniquely mapped to its four 16-bit entries. See [EQADC result FIFOs](#) for more information on RFIFOs. These registers are read only. Data written to these registers is ignored.

Address: 0h base + 3C0h offset + (4d × i), where i=0d to 3d

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16           | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | RFIFO3_DATAW |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0            | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

#### EQADC\_RF3Rn field descriptions

| Field                 | Description                                                                                                                                                                                                                                |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>Reserved      | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                    |
| 16–31<br>RFIFO3_DATAW | RFIFO3 Data w (w = 0, ..., 3)<br>Reading RFIFOx_DATAw returns the value stored on the wth entry of RFIFOx. Each RFIFO is composed of four 16-bit entries, with register 0 being mapped to the one with the smallest memory mapped address. |

### 29.5.31 EQADC RFIFO4 Register (EQADC\_RF4Rn)

The EQADC RFIFO Registers (EQADC\_RFxRw) (x=0, ..., 5; w=0, ..., 3) provide visibility of the contents of a RFIFO for debugging purposes. Each RFIFO has four registers which are uniquely mapped to its four 16-bit entries. See [EQADC result FIFOs](#) for more information on RFIFOs. These registers are read only. Data written to these registers is ignored.

Address: 0h base + 400h offset + (4d × i), where i=0d to 3d

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16           | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | RFIFO4_DATAW |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0            | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**EQADC\_RF4Rn field descriptions**

| Field                 | Description                                                                                                                                                                                                                                |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>Reserved      | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                    |
| 16–31<br>RFIFO4_DATAW | RFIFO4 Data w (w = 0, ..., 3)<br>Reading RFIFOx_DATAw returns the value stored on the wth entry of RFIFOx. Each RFIFO is composed of four 16-bit entries, with register 0 being mapped to the one with the smallest memory mapped address. |

**29.5.32 EQADC RFIFO5 Register (EQADC\_RF5Rn)**

The EQADC RFIFO Registers (EQADC\_RFxRw) (x=0, ..., 5; w=0, ..., 3) provide visibility of the contents of a RFIFO for debugging purposes. Each RFIFO has four registers which are uniquely mapped to its four 16-bit entries. See [EQADC result FIFOs](#) for more information on RFIFOs. These registers are read only. Data written to these registers is ignored.

Address: 0h base + 440h offset + (4d × i), where i=0d to 3d

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16           | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | RFIFO5_DATAW |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0            | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**EQADC\_RF5Rn field descriptions**

| Field                 | Description                                                                                                                                                                                                                                |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>Reserved      | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                    |
| 16–31<br>RFIFO5_DATAW | RFIFO5 Data w (w = 0, ..., 3)<br>Reading RFIFOx_DATAw returns the value stored on the wth entry of RFIFOx. Each RFIFO is composed of four 16-bit entries, with register 0 being mapped to the one with the smallest memory mapped address. |

**29.5.33 On-chip ADC registers**

This section describes a list of registers that control on-chip ADC operation. The ADC registers are not part of the CPU accessible memory map. These registers can only be accessed indirectly through configuration commands. There are non memory mapped registers that are individual per ADC, plus other registers that are shared by both ADCs. The address, usage, and access privilege of each register is shown in [Table 29-110](#). Data written to or read from reserved areas of the memory map is undefined.

Their assigned addresses are the values used to set the ADC\_REG\_ADDRESS field of the read/write configurations commands bound for the on-chip ADCs. These are half-word addresses. Further, the following restrictions apply when accessing these registers:

- Registers ADC0\_CR, ADC0\_GCCR, ADC0\_OCCR, ADC0\_AGR1/2 and ADC0\_AOR1/2 can only be accessed by configuration commands sent to CBuffer0.
- Registers ADC1\_CR, ADC1\_GCCR, ADC1\_OCCR, ADC1\_AGR1/2 and ADC1\_AOR1/2 can only be accessed by configuration commands sent to CBuffer1.
- Registers ADC\_TSCR, ADC\_TBCR, ADC\_ACR1-14, ADC\_EACR1-14 and ADC\_PUDCR0-7 can be accessed by configuration commands sent to CBuffer0 or to CBuffer1. A data write to any of these registers through a configuration command sent to CBuffer0 will write the same memory location as when writing to it through a configuration command sent to CBuffer1.

### Note

Simultaneous write accesses from CBuffer0 and CBuffer1 to any of the shared registers are not allowed.

**Table 29-110. On-Chip ADC Memory Map**

| ADC address | Use                                                                                                                                                  | Access     |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| 0x00        | ADC0/ADC1 <sup>1</sup> Conversion Command for Standard Configuration (See <a href="#">Conversion command format for the standard configuration</a> ) | Write      |
| 0x01        | ADC0/ADC1 Configuration Control Register (ADC0_CR, ADC1_CR)                                                                                          | Write/Read |
| 0x02        | Time Stamp Control Register (ADC_TSCR)                                                                                                               | Write/Read |
| 0x03        | Time Base Counter Register (ADC_TBCR)                                                                                                                | Write/Read |
| 0x04        | ADC0/ADC1 Gain Calibration Constant Register (ADC0_GCCR, ADC1_GCCR)                                                                                  | Write/Read |
| 0x05        | ADC0/ADC1 Offset Calibration Constant Register (ADC0_OCCR, ADC1_OCCR)                                                                                | Write/Read |
| 0x06- 0x07  | Reserved                                                                                                                                             | -          |
| 0x08        | ADC0/ADC1 Conversion Command for Alternate Configuration 1 (See <a href="#">Conversion command format for alternate configurations</a> )             | Write      |
| 0x09        | ADC0/ADC1 Conversion Command for Alternate Configuration 2 (See <a href="#">Conversion command format for alternate configurations</a> )             | Write      |
| 0x0A        | ADC0/ADC1 Conversion Command for Alternate Configuration 3 (See <a href="#">Conversion command format for alternate configurations</a> )             | Write      |
| 0x0B        | ADC0/ADC1 Conversion Command for Alternate Configuration 4 (See <a href="#">Conversion command format for alternate configurations</a> )             | Write      |
| 0x0C        | ADC0/ADC1 Conversion Command for Alternate Configuration 5 (See <a href="#">Conversion command format for alternate configurations</a> )             | Write      |
| 0x0D        | ADC0/ADC1 Conversion Command for Alternate Configuration 6 (See <a href="#">Conversion command format for alternate configurations</a> )             | Write      |

*Table continues on the next page...*

**Table 29-110. On-Chip ADC Memory Map (continued)**

| ADC address | Use                                                                                                                                          | Access     |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------|------------|
| 0x0E        | ADC0/ADC1 Conversion Command for Alternate Configuration 7<br>(See <a href="#">Conversion command format for alternate configurations</a> )  | Write      |
| 0x0F        | ADC0/ADC1 Conversion Command for Alternate Configuration 8<br>(See <a href="#">Conversion command format for alternate configurations</a> )  | Write      |
| 0x10        | ADC0/ADC1 Conversion Command for Alternate Configuration 9<br>(See <a href="#">Conversion command format for alternate configurations</a> )  | Write      |
| 0x11        | ADC0/ADC1 Conversion Command for Alternate Configuration 10<br>(See <a href="#">Conversion command format for alternate configurations</a> ) | Write      |
| 0x12        | ADC0/ADC1 Conversion Command for Alternate Configuration 11<br>(See <a href="#">Conversion command format for alternate configurations</a> ) | Write      |
| 0x13        | ADC0/ADC1 Conversion Command for Alternate Configuration 12<br>(See <a href="#">Conversion command format for alternate configurations</a> ) | Write      |
| 0x14        | ADC0/ADC1 Conversion Command for Alternate Configuration 13<br>(See <a href="#">Conversion command format for alternate configurations</a> ) | Write      |
| 0x15        | ADC0/ADC1 Conversion Command for Alternate Configuration 14<br>(See <a href="#">Conversion command format for alternate configurations</a> ) | Write      |
| 0x16-0x2F   | Reserved                                                                                                                                     | -          |
| 0x30        | Alternate Configuration 1 Control Register (ADC_ACR1)                                                                                        | Write/Read |
| 0x31        | ADC0/ADC1 Alternate Gain 1 Register (ADC0_AGR1,<br>ADC1_AGR1)                                                                                | Write/Read |
| 0x32        | ADC0/ADC1 Alternate Offset 1 Register (ADC0_AOR1,<br>ADC1_AOR1)                                                                              | Write/Read |
| 0x33        | Extended Alternate Configuration 1 Control Register (ADC_EACR1)                                                                              | Write/Read |
| 0x34        | Alternate Configuration 2 Control Register (ADC_ACR2)                                                                                        | Write/Read |
| 0x35        | ADC0/ADC1 Alternate Gain 2 Register (ADC0_AGR2,<br>ADC1_AGR2)                                                                                | Write/Read |
| 0x36        | ADC0/ADC1 Alternate Offset 2 Register (ADC0_AOR2,<br>ADC1_AOR2)                                                                              | Write/Read |
| 0x37        | Extended Alternate Configuration 2 Control Register (ADC_EACR2)                                                                              | Write/Read |
| 0x38        | Alternate Configuration 3 Control Register (ADC_ACR3)                                                                                        | Write/Read |
| 0x39        | Reserved                                                                                                                                     | -          |
| 0x3A        | Reserved                                                                                                                                     | -          |
| 0x3B        | Extended Alternate Configuration 3 Control Register (ADC_EACR3)                                                                              | Write/Read |
| 0x3C        | Alternate Configuration 4 Control Register (ADC_ACR4)                                                                                        | Write/Read |
| 0x3D        | Reserved                                                                                                                                     | -          |
| 0x3E        | Reserved                                                                                                                                     | -          |
| 0x3F        | Extended Alternate Configuration 4 Control Register (ADC_EACR4)                                                                              | Write/Read |
| 0x40        | Alternate Configuration 5 Control Register (ADC_ACR5)                                                                                        | Write/Read |
| 0x41        | Reserved                                                                                                                                     | -          |
| 0x42        | Reserved                                                                                                                                     | -          |
| 0x43        | Extended Alternate Configuration 5 Control Register (ADC_EACR5)                                                                              | Write/Read |
| 0x44        | Alternate Configuration 6 Control Register (ADC_ACR6)                                                                                        | Write/Read |
| 0x45        | Reserved                                                                                                                                     | -          |

*Table continues on the next page...*

**Table 29-110. On-Chip ADC Memory Map (continued)**

| ADC address | Use                                                               | Access     |
|-------------|-------------------------------------------------------------------|------------|
| 0x46        | Reserved                                                          | -          |
| 0x47        | Extended Alternate Configuration 6 Control Register (ADC_EACR6)   | Write/Read |
| 0x48        | Alternate Configuration 7 Control Register (ADC_ACR7)             | Write/Read |
| 0x49        | Reserved                                                          | -          |
| 0x4A        | Reserved                                                          | -          |
| 0x4B        | Extended Alternate Configuration 7 Control Register (ADC_EACR7)   | Write/Read |
| 0x4C        | Alternate Configuration 8 Control Register (ADC_ACR8)             | Write/Read |
| 0x4D        | Reserved                                                          | -          |
| 0x4E        | Reserved                                                          | -          |
| 0x4F        | Extended Alternate Configuration 8 Control Register (ADC_EACR8)   | Write/Read |
| 0x50        | Alternate Configuration 9 Control Register (ADC_ACR9)             | Write/Read |
| 0x51        | Reserved                                                          | -          |
| 0x52        | Reserved                                                          | -          |
| 0x53        | Extended Alternate Configuration 9 Control Register (ADC_EACR9)   | Write/Read |
| 0x54        | Alternate Configuration 10 Control Register (ADC_ACR10)           | Write/Read |
| 0x55        | Reserved                                                          | -          |
| 0x56        | Reserved                                                          | -          |
| 0x57        | Extended Alternate Configuration 10 Control Register (ADC_EACR10) | Write/Read |
| 0x58        | Alternate Configuration 11 Control Register (ADC_ACR11)           | Write/Read |
| 0x59        | Reserved                                                          | -          |
| 0x5A        | Reserved                                                          | -          |
| 0x5B        | Extended Alternate Configuration 11 Control Register (ADC_EACR11) | Write/Read |
| 0x5C        | Alternate Configuration 12 Control Register (ADC_ACR12)           | Write/Read |
| 0x5D        | Reserved                                                          | -          |
| 0x5E        | Reserved                                                          | -          |
| 0x5F        | Extended Alternate Configuration 12 Control Register (ADC_EACR12) | Write/Read |
| 0x60        | Alternate Configuration 13 Control Register (ADC_ACR13)           | Write/Read |
| 0x61        | Reserved                                                          | -          |
| 0x62        | Reserved                                                          | -          |
| 0x63        | Extended Alternate Configuration 13 Control Register (ADC_EACR13) | Write/Read |
| 0x64        | Alternate Configuration 14 Control Register (ADC_ACR14)           | Write/Read |
| 0x65        | Reserved                                                          | -          |
| 0x66        | Reserved                                                          | -          |
| 0x67        | Extended Alternate Configuration 14 Control Register (ADC_EACR14) | Write/Read |
| 0x68-0x6F   | Reserved                                                          | -          |
| 0x70        | Pull Up/Down Control Register0 (ADC_PUDCR0)                       | Write/Read |

*Table continues on the next page...*

**Table 29-110. On-Chip ADC Memory Map (continued)**

| ADC address | Use                                                                            | Access     |
|-------------|--------------------------------------------------------------------------------|------------|
| 0x71        | Pull Up/Down Control Register1 (ADC_PUDCR1)                                    | Write/Read |
| 0x72        | Pull Up/Down Control Register2 (ADC_PUDCR2)                                    | Write/Read |
| 0x73        | Pull Up/Down Control Register3 (ADC_PUDCR3)                                    | Write/Read |
| 0x74        | Pull Up/Down Control Register4 (ADC_PUDCR4)                                    | Write/Read |
| 0x75        | Pull Up/Down Control Register5 (ADC_PUDCR5)                                    | Write/Read |
| 0x76        | Pull Up/Down Control Register6 (ADC_PUDCR6)                                    | Write/Read |
| 0x77        | Pull Up/Down Control Register7 (ADC_PUDCR7)                                    | Write/Read |
| 0x78-0x97   | Pull Up/Down Control Register 8 to 39 (ADC_PUDCR8 to ADC_PUDCR39) <sup>2</sup> | Write/Read |
| 0x98-0x9F   | Reserved                                                                       | -          |
| 0xA0        | STAC Bus Time Base 1 Register (ADC_STACTBR1)                                   | Read       |
| 0xA1        | STAC Bus Time Base 2 Register (ADC_STACTBR2)                                   | Read       |
| 0xA2-0xFE   | Reserved                                                                       | -          |
| 0xFF        | Reserved for dummy write command <sup>3</sup>                                  | Write      |

1. Throughout the table, ADC0/ADC1 indicates that if the command is stored in CBuffer0 it will be applied to ADC0 and if in CBuffer1 it applies to ADC1. If this indication is omitted the register applies for both ADC0 and ADC1, independent of the CBuffer used.
2. See the chip-specific EQADC information for details about which analog inputs have pull up/pull down control
3. Defined as an address for dummy write operation. Read is not allowed.

### 29.5.33.1 ADC0/1 Control Registers (ADC0\_CR and ADC1\_CR)

The ADC0/1 Control Registers (ADC0/1\_CR) is used to define the standard configuration of the ADC. In the standard configuration, the parameters contained in the Alternate Configuration Control Registers (ADC\_ACR1-14) are fixed at their reset value. A conversion uses the standard configuration when the conversion command (with the standard format) is written to address 0x00 of the on-chip ADC memory map. Refer to [Conversion command format for the standard configuration](#).

## Memory map and register definition

ADC0 Register address: 0x01

|        | 0       | 1 | 2 | 3 | 4         | 5 | 6          | 7           | 8            | 9            | 10          | 11 | 12 | 13 | 14 | 15 |
|--------|---------|---|---|---|-----------|---|------------|-------------|--------------|--------------|-------------|----|----|----|----|----|
| R      | ADC0_EN | 0 | 0 | 0 | ADC0_EMUX | 0 | ADC0_TBSEL | ADC0_ODD_PS | ADC0_CLK_DTY | ADC0_CLK_SEL | ADC0_CLK_PS |    |    |    |    |    |
| W      |         |   |   |   | X         |   |            |             |              |              |             |    |    |    |    |    |
| RESET: | 0       | 0 | 0 | 0 | 0         | 0 | 0          | 0           | 0            | 0            | 0           | 1  | 1  | 1  | 1  | 1  |

ADC1 Register address: 0x01

|        | 0       | 1 | 2 | 3 | 4         | 5 | 6          | 7           | 8            | 9            | 10          | 11 | 12 | 13 | 14 | 15 |
|--------|---------|---|---|---|-----------|---|------------|-------------|--------------|--------------|-------------|----|----|----|----|----|
| R      | ADC1_EN | 0 | 0 | 0 | ADC1_EMUX | 0 | ADC1_TBSEL | ADC1_ODD_PS | ADC1_CLK_DTY | ADC1_CLK_SEL | ADC1_CLK_PS |    |    |    |    |    |
| W      |         |   |   |   | X         |   |            |             |              |              |             |    |    |    |    |    |
| RESET: | 0       | 0 | 0 | 0 | 0         | 0 | 0          | 0           | 0            | 0            | 0           | 1  | 1  | 1  | 1  | 1  |

 = Unimplemented or Reserved

**Figure 29-104. ADC0/1 Control Registers (ADC0/1\_CR)**

**Table 29-111. ADC0/1\_CR field descriptions**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ADC0/1_EN         | <p>Enable bit for ADC0/1</p> <p>ADC0/1_EN enables ADC0/1 to perform A/D conversions. See <a href="#">Enabling and disabling the on-chip ADCs</a> for details.</p> <p><b>Note:</b> The bias generator circuit inside the ADC hard macro ceases functioning when both ADC0_EN and ADC1_EN bits are negated.</p> <p><b>Note:</b> Conversion commands sent to the CBuffer of a disabled ADC are ignored by the ADC control hardware.</p> <p><b>Note:</b> When the ADC0/1_EN status is changed from asserted to negated, the ADC Clock will not stop until it reaches its low phase.</p> <p>0                      ADC is disabled. Clock supply to ADC0/1 is stopped.</p> <p>1                      ADC is enabled and ready to perform A/D conversions.</p>                                                                                                       |
| ADC0/1_EMUX       | <p>External Multiplexer enable for ADC0/1</p> <p>When ADC0/1_EMUX is asserted, the MA pins will output digital values according to the number of the external channel being converted for selecting external multiplexer inputs. See <a href="#">Internal/external multiplexing</a> for a detailed description about how ADC0/1_EMUX affects channel number decoding.</p> <p><b>Note:</b> Both ADC0/1_EMUX bits must not be asserted at the same time.</p> <p><b>Note:</b> The ADC0/1_EMUX bit must only be written when the ADC0/1_EN bit is negated. ADC0/1_EMUX can be set during the same write cycle used to set ADC0/1_EN.</p> <p>0                      External multiplexer disabled; no external multiplexer channels can be selected.</p> <p>1                      External multiplexer enabled; external multiplexer channels can be selected.</p> |
| ADC0/1_TBSEL[0:1] | Timebase Selection for ADC0/1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

*Table continues on the next page...*



**Table 29-111. ADC0/1\_CR field descriptions (continued)**

| Field              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                    | <p>The ADC0/1_TBSEL[0:1] field selects the time information to be used as timestamp according to <a href="#">Table 29-112</a>.</p> <p><b>Note:</b> This selection is overridden by the corresponding field ATBSEL in the ADC_ACR1-14 registers when the alternate conversion command is used.</p>                                                                                                                                                                                                                                                                                                                                                                   |
| ADC0/1_ODD_PS      | <p>Clock Prescaler Odd Rates Selector for ADC0/1</p> <p>The ADC0/1_CLK_DTY field controls the duty rate of the ADC0/1 clock when the ADC0/1_CLK_PS field is asserted. The generated clock has an odd number of system clock cycles, therefore this field is used to select a clock duty higher or lower than 50%.</p> <p>0 Even divide factor is selected. The final divide factor is dependent of ADC0/1_CLK_PS field.</p> <p>1 Odd divide factor is selected. The final divide factor is dependent of ADC0/1_CLK_PS field.</p>                                                                                                                                    |
| ADC0/1_CLK_DTY     | <p>Clock Duty Rate Selector for ADC0/1 (for odd divide factors)</p> <p>The ADC0/1_ODD_PS field is used together with the ADC0/1_CLK_PS field to define even/odd divide factors in the generation of the ADC0/1 clocks. See <a href="#">Table 29-113</a> for available divide factors.</p> <p>0 clock low interval is longer 1 clock cycle than high pulse.</p> <p>1 clock high pulse is longer 1 clock cycle than low portion.</p>                                                                                                                                                                                                                                  |
| ADC0/1_CLK_SEL     | <p>Clock Selector for ADC0/1</p> <p>The ADC0/1_CLK_SEL is used to select between the system clock signal or the prescaler output signal. The prescaler provides the system clock signal divided by a even factor from 2 to 64. This is required to permit the ADC to run as fast as possible when the device is in Low Power Active mode and system clock is around 1 MHz.</p> <p><b>Note:</b> The ADC0/1_CLK_SEL bits must only be written when the ADC0/1_EN bit is negated. ADC0/1_CLK_SEL can be set during the same write cycle used to set ADC0/1_EN.</p> <p>0 Prescaler output clock is selected.</p> <p>1 System clock is selected - maximum frequency.</p> |
| ADC0/1_CLK_PS[0:4] | <p>Clock Prescaler Field for ADC0/1</p> <p>The ADC0/1_CLK_PS field controls the system clock divide factor for the ADC0/1 clock as in <a href="#">Table 29-113</a>. See <a href="#">ADC clock and conversion speed</a> for details about how to set ADC0/1_CLK_PS.</p> <p><b>Note:</b> The ADC0/1_CLK_PS field must only be written when the ADC0/1_EN bit is negated. This field can be configured during the same write cycle used to set ADC0/1_EN.</p>                                                                                                                                                                                                          |

**Table 29-112. Timebase selection**

| ADC0/1_TBSEL[0:1] | Definition                                                                          |
|-------------------|-------------------------------------------------------------------------------------|
| 00                | Selects internally generated time base as time stamp.                               |
| 01                | Selects imported time base 1 indicated by SRV1 bit field of EQADC_STACCCR register. |
| 10                | Selects imported time base 2 indicated by SRV2 bit field of EQADC_STACCCR register. |
| 11                | Reserved                                                                            |

**Table 29-113. System Clock Divide Factor for ADC Clock**

| ADC0/1_CLK_PS[0:4] | System Clock Divide Factor |                   |
|--------------------|----------------------------|-------------------|
|                    | ADC0/1_ODD_PS = 0          | ADC0/1_ODD_PS = 1 |
| 0b00000            | 2                          | 3                 |
| 0b00001            | 4                          | 5                 |
| 0b00010            | 6                          | 7                 |
| 0b00011            | 8                          | 9                 |
| 0b00100            | 10                         | 11                |
| 0b00101            | 12                         | 13                |
| 0b00110            | 14                         | 15                |
| 0b00111            | 16                         | 17                |
| 0b01000            | 18                         | 19                |
| 0b01001            | 20                         | 21                |
| 0b01010            | 22                         | 23                |
| 0b01011            | 24                         | 25                |
| 0b01100            | 26                         | 27                |
| 0b01101            | 28                         | 29                |
| 0b01110            | 30                         | 31                |
| 0b01111            | 32                         | 33                |
| 0b10000            | 34                         | 35                |
| 0b10001            | 36                         | 37                |
| 0b10010            | 38                         | 39                |
| 0b10011            | 40                         | 41                |
| 0b10100            | 42                         | 43                |
| 0b10101            | 44                         | 45                |
| 0b10110            | 46                         | 47                |
| 0b10111            | 48                         | 49                |
| 0b11000            | 50                         | 51                |
| 0b11001            | 52                         | 53                |
| 0b11010            | 54                         | 55                |
| 0b11011            | 56                         | 57                |
| 0b11100            | 58                         | 59                |
| 0b11101            | 60                         | 61                |
| 0b11110            | 62                         | 63                |
| 0b11111            | 64                         | 65                |

### 29.5.33.2 ADC Time Stamp Control Register (ADC\_TSCR)

The ADC Time Stamp Control Register (ADC\_TSCR) contains a system clock divide factor used in the making of the time base counter clock. It determines at what frequency the time base counter will run. ADC\_TSCR can be accessed by configuration commands sent to CBuffer0 or to CBuffer1. A data write to ADC\_TSCR through a configuration command sent to CBuffer0 will write the same memory location as when writing to it through a configuration command sent to CBuffer1.

#### Note

Simultaneous write accesses from CBuffer0 and CBuffer1 to ADC\_TSCR are not allowed.

ADC0/1 Register address: 0x02

|        | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12         | 13 | 14 | 15 |
|--------|---|---|---|---|---|---|---|---|---|---|----|----|------------|----|----|----|
| R      | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | TBC_CLK_PS |    |    |    |
| W      |   |   |   |   |   |   |   |   |   |   |    |    |            |    |    |    |
| RESET: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0          | 0  | 0  | 0  |

 = Unimplemented or Reserved

**Figure 29-105. ADC Time Stamp Control Register (ADC\_TSCR)**

**Table 29-114. ADC\_TSCR field descriptions**

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TBC_CLK_PS[0:3] | <p>Time Base Counter Clock Prescaler</p> <p>The TBC_CLK_PS field contains the system clock divide factor for the time base counter. It controls the accuracy of the time stamp. The prescaler is disabled when TBC_CLK_PS is set to 0b0000. See <a href="#">Table 29-115</a>.</p> <p><b>Note:</b> If TBC_CLK_PS is not set to disabled, it must not be changed to any other value besides disabled. If TBC_CLK_PS is set to disabled it can be changed to any other value.</p> |

**Table 29-115. Clock divide factor for time stamp**

| TBC_CLK_PS[0:3] | System clock divide factor | Clock to time stamp counter for a 120 MHz system clock (MHz) |
|-----------------|----------------------------|--------------------------------------------------------------|
| 0b0000          | Disabled                   | Disabled                                                     |
| 0b0001          | 1                          | 120                                                          |
| 0b0010          | 2                          | 60                                                           |
| 0b0011          | 4                          | 30                                                           |
| 0b0100          | 6                          | 20                                                           |
| 0b0101          | 8                          | 15                                                           |

*Table continues on the next page...*

**Table 29-115. Clock divide factor for time stamp (continued)**

| TBC_CLK_PS[0:3] | System clock divide factor | Clock to time stamp counter for a 120 MHz system clock (MHz) |
|-----------------|----------------------------|--------------------------------------------------------------|
| 0b0110          | 10                         | 12                                                           |
| 0b0111          | 12                         | 10                                                           |
| 0b1000          | 16                         | 7.5                                                          |
| 0b1001          | 32                         | 3.75                                                         |
| 0b1010          | 64                         | 1.88                                                         |
| 0b1011          | 128                        | 0.94                                                         |
| 0b1100          | 256                        | 0.47                                                         |
| 0b1101          | 512                        | 0.23                                                         |
| 0b1110 - 0b1111 | Reserved                   | -                                                            |

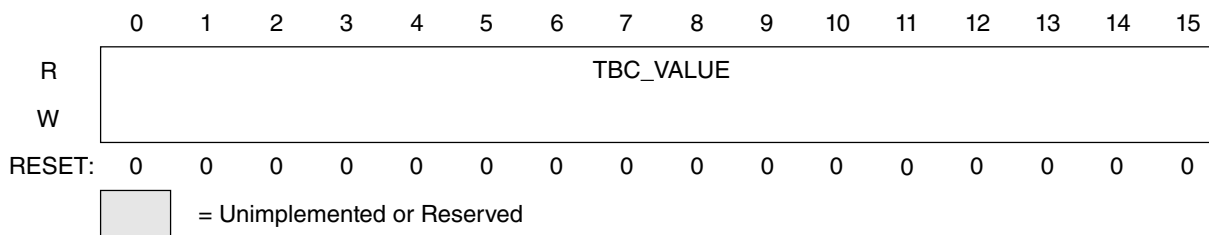
### 29.5.33.3 ADC Time Base Counter Registers (ADC\_TBCR)

The ADC Time Base Counter Register (ADC\_TBCR) contains the current value of the time base counter. ADC\_TBCR can be accessed by configuration commands sent to CBuffer0 or to CBuffer1. A data write to ADC\_TBCR through a configuration command sent to CBuffer0 will write the same memory location as when writing to it through a configuration command sent to CBuffer1.

#### Note

Simultaneous write accesses from CBuffer0 and CBuffer1 to ADC\_TBCR are not allowed.

ADC0/1 Register address: 0x03

**Figure 29-106. ADC Time Base Counter Register (ADC\_TBCR)****Table 29-116. ADC\_TBCR field descriptions**

| Field           | Description                   |
|-----------------|-------------------------------|
| TBC_VALUE[0:15] | Time Base Counter VALUE Field |

**Table 29-116. ADC\_TBCR field descriptions**

| Field | Description                                                                                                                                                                                                                                                                                      |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | The TBC_VALUE field contains the current value of the time base counter. Reading TBC_VALUE returns the current value of time base counter. Writes to TBC_VALUE register load the written data to the counter. The time base counter counts from 0x0000 to 0xFFFF and wraps when reaching 0xFFFF. |

### 29.5.33.4 ADC0/1 Gain Calibration Constant Registers (ADC0\_GCCR and ADC1\_GCCR)

The ADC0/1 Gain Calibration Constant Register (ADC0/1\_GCCR) contains the gain calibration constant used to fine-tune the ADC0/1 conversion results. See [ADC calibration feature](#) for details about the calibration scheme used in the EQADC.

**ADC0 Register address: 0x04**

|        |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|--------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
|        | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R      | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W      |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| RESET: | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

**ADC1 Register address: 0x04**

|        |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|--------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
|        | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R      | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W      |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| RESET: | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |
|        |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |

= Unimplemented or Reserved

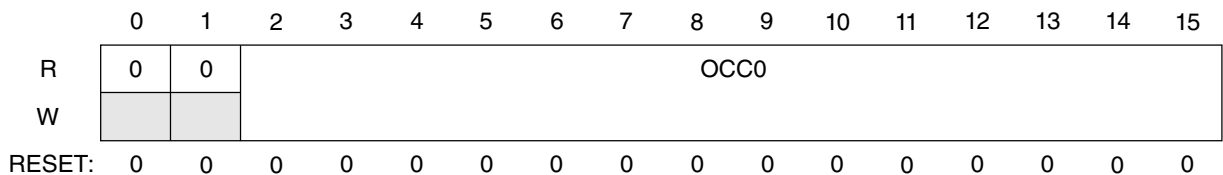
**Figure 29-107. ADC0/1 Gain Calibration Constant Registers (ADC0/1\_GCCR)****Table 29-117. ADC0/1\_GCCR field descriptions**

| Field        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GCC0/1[0:14] | Gain calibration constant for ADC0/1<br><br>GCC0/1 contains the gain calibration constant used to fine-tune ADC0/1 conversion results. It is a unsigned 15-bit fixed pointed value. The gain calibration constant is an unsigned fixed point number expressed in the GCC_INT.GCC_FRAC binary format. The integer part of the gain constant (GCC_INT) contains a single binary digit while its fractional part (GCC_FRAC) contains fourteen digits. For details about the GCC data format, see <a href="#">MAC unit and operand data format</a> . |

29.5.33.5    **ADC0/1 Offset Calibration Constant Registers  
(ADC0\_OCCR and ADC1\_OCCR)**

The ADC0/1 Offset Calibration Constant Register (ADC0/1\_OCCR) contains the offset calibration constant used to fine-tune of ADC0/1 conversion results. The offset constant is a signed 14-bit integer value. See [ADC calibration feature](#) for details about the calibration scheme used in the EQADC.

**ADC0 Register address: 0x05**



**ADC1 Register address: 0x05**

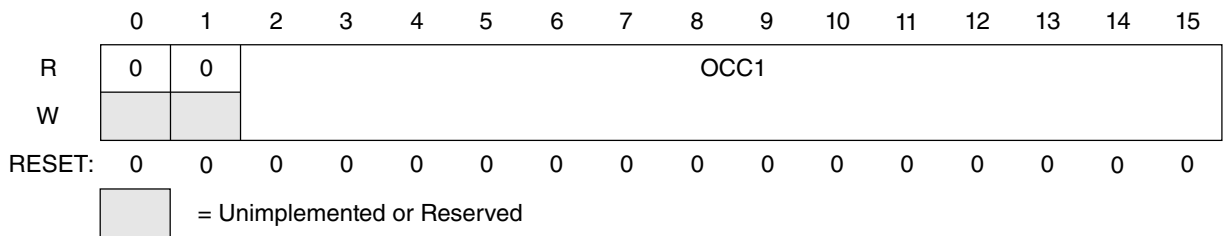


**Figure 29-108. ADC0/1 Offset Calibration Constant Registers (ADC0/1\_OCCR)**

**Table 29-118. ADC0/1\_OCCR field descriptions**

| Field        | Description                                                                                                                                                                                                              |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| OCC0/1[0:13] | Offset Calibration Constant of ADC0/1<br><br>OCC0/1 contains the offset calibration constant used to fine-tune ADC0/1 conversion results. Negative values should be expressed using the two's complement representation. |

29.5.33.6    **Alternate Configuration 1-14 Control Registers  
(ADC\_ACR1-14)**

The Alternate Configuration Control Registers (ADC\_ACR1-14) are used to configure the alternate configurations of the ADC. There are 14 possible alternate configurations, each one associated with one of the ADC\_ACR1-14 registers. All alternate configurations share the same standard configuration parameters from the ADC0/1\_CR

registers, plus additional configuration parameters contained in the ADC\_ACR1-14. A conversion uses one of the alternate configurations when the conversion command (with the alternate configuration format) is written to an address in the range 0x08-0x15 of the on-chip ADC memory map. It also can use the Extended Alternated Configurations as described in [Extended Alternate Configuration 1-14 Control Registers \(ADC\\_EACR1-14\)](#) if bit DAM=1. See [Conversion command format for alternate configurations](#).

**ADC0/1 Register address: 0x30**

ADC0/1 Register address: 0x34

ADC0/1 Register address: 0x38

ADC0/1 Register address: 0x3C

ADC0/1 Register address: 0x40

ADC0/1 Register address: 0x44

ADC0/1 Register address: 0x48

ADC0/1 Register address: 0x4C

ADC0/1 Register address: 0x50

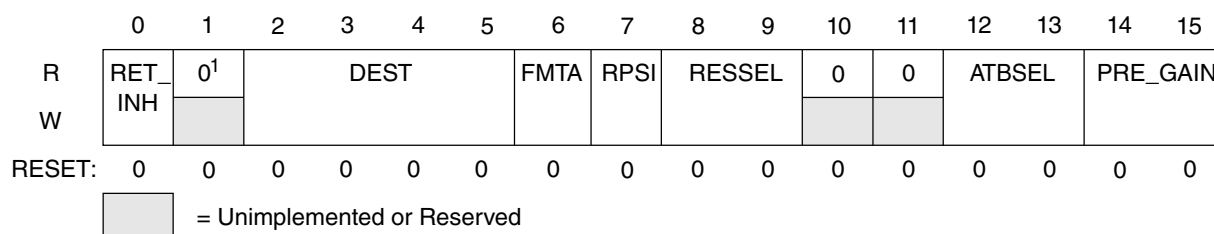
ADC0/1 Register address: 0x54

ADC0/1 Register address: 0x58

ADC0/1 Register address: 0x5C

ADC0/1 Register address: 0x60

ADC0/1 Register address: 0x64



<sup>1</sup> Reserved for expansion of DEST.

**Figure 29-109. Alternate Configuration 1-14 Control Registers (ADC\_ACR1-14)**

**Table 29-119. ADC\_ACR1-14 field descriptions**

| Field     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RET_INH   | <p>Result Transfer Inhibit / Decimation Filter Pre-Fill</p> <p>This bit is used to inhibit the transfer of the result data from the peripheral module to the result queue. When the module is a Decimation Filter, this bit sets the filter in a special mode (PRE-FILL) in which it does not generate decimated samples out from the conversion results received from the EQADC block, but the conversion samples are used by the filter algorithm. This feature allows a proper initialization of the Decimation Filter without generating any decimated result. Or this bit is useful for sending the result of the ADC to the STAC bus master but not putting the result in the result queue.</p> <p><b>Note:</b> Refer to the block guide of each peripheral module to obtain the correct behavior of the companion module to the PRE-FILL mode.</p> <p>0 Result transfer to result queue / Decimation Filter in filtering mode</p> <p>1 No result transfer to result queue / Decimation Filter PRE-FILL mode</p> |
| DEST[0:3] | Conversion Result Destination Selection                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

*Table continues on the next page...*

**Table 29-119. ADC\_ACR1-14 field descriptions (continued)**

| Field         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |    |                                                                                              |    |                                                                                     |    |                                                                                     |    |          |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|----------------------------------------------------------------------------------------------|----|-------------------------------------------------------------------------------------|----|-------------------------------------------------------------------------------------|----|----------|
|               | <p>The DEST[0:3] field selects the primary destination of the conversion result generated by the Alternate Conversion Command as shown in <a href="#">Table 29-120</a>. This field also affects the behavior of the FMTA bit and the FFMT bit of the conversion command for alternate configurations (see <a href="#">Conversion command format for alternate configurations</a>). And this field is affected by DAM bit of the EQADC Module Configuration Register (see <a href="#">EQADC Module Configuration Register (EQADC_MCR)</a>). There is also configurations to send the conversion result to a second destination using the Extended Alternate Configuration registers (<a href="#">Extended Alternate Configuration 1-14 Control Registers (ADC_EACR1-14)</a>).</p>                                                                                                                        |    |                                                                                              |    |                                                                                     |    |                                                                                     |    |          |
| FMTA          | <p>Conversion Data Format for Alternate Configuration</p> <p>If the DEST field is not 0b000, the FMTA bit specifies how the 12-bit conversion data returned by the ADCs is formatted into the 16-bit data which is sent to the parallel side interface.</p> <table> <tr> <td>0</td><td>Right justified unsigned</td></tr> <tr> <td>1</td><td>Right justified signed</td></tr> </table>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 0  | Right justified unsigned                                                                     | 1  | Right justified signed                                                              |    |                                                                                     |    |          |
| 0             | Right justified unsigned                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |    |                                                                                              |    |                                                                                     |    |                                                                                     |    |          |
| 1             | Right justified signed                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |    |                                                                                              |    |                                                                                     |    |                                                                                     |    |          |
| RPSI          | <p>Return Parallel Side Interface</p> <p>The RPSI bit is used in the companion modules that have two PSI interfaces (PSI0 and PSI1), which are connected to two eQADCs. It specifies in the companion module if the data received from one PSI should be transmitted to the same PSI or to the other one.</p> <p><b>Note:</b> The companion module may take samples from two eQADCs, allowing double sample rate. If the RPSI bits are different, the results are sent to just one of the eQADCs.</p> <p><b>Note:</b> It is not allowed to use RPSI=1 when the SoC has only one eQADC. It can occur errors in the companion module due to routing error.</p> <table> <tr> <td>0</td><td>Companion module results from this eQADC are sent back to the same eQADC using the same PSI.</td></tr> <tr> <td>1</td><td>Companion module results from this eQADC are sent to the other PSI</td></tr> </table> | 0  | Companion module results from this eQADC are sent back to the same eQADC using the same PSI. | 1  | Companion module results from this eQADC are sent to the other PSI                  |    |                                                                                     |    |          |
| 0             | Companion module results from this eQADC are sent back to the same eQADC using the same PSI.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |    |                                                                                              |    |                                                                                     |    |                                                                                     |    |          |
| 1             | Companion module results from this eQADC are sent to the other PSI                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |    |                                                                                              |    |                                                                                     |    |                                                                                     |    |          |
| RESSEL[0:1]   | <p>ADC Resolution Selection</p> <p>The RESSEL[0:1] field selects the resolution of the ADC.</p> <table> <tr> <td>00</td><td>ADC set to 12-bits resolution</td></tr> <tr> <td>01</td><td>ADC set to 10-bits resolution</td></tr> <tr> <td>10</td><td>ADC set to 8-bits resolution</td></tr> <tr> <td>11</td><td>Reserved</td></tr> </table>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | 00 | ADC set to 12-bits resolution                                                                | 01 | ADC set to 10-bits resolution                                                       | 10 | ADC set to 8-bits resolution                                                        | 11 | Reserved |
| 00            | ADC set to 12-bits resolution                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |    |                                                                                              |    |                                                                                     |    |                                                                                     |    |          |
| 01            | ADC set to 10-bits resolution                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |    |                                                                                              |    |                                                                                     |    |                                                                                     |    |          |
| 10            | ADC set to 8-bits resolution                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |    |                                                                                              |    |                                                                                     |    |                                                                                     |    |          |
| 11            | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |    |                                                                                              |    |                                                                                     |    |                                                                                     |    |          |
| ATBSEL[0:1]   | <p>Alternate Command Timebase Selector</p> <p>The ATBSEL[0:1] field selects the time information to be used as timestamp</p> <p><b>Note:</b> This selection overrides the corresponding fields ADC0/1_TBSEL in the ADC0/1_CR registers when the alternate conversion command is used.</p> <table> <tr> <td>00</td><td>Selects internally generated time base as time stamp</td></tr> <tr> <td>01</td><td>Selects imported time base 1 indicated by SRV1 bit field of EQADC_STACCCR register.</td></tr> <tr> <td>10</td><td>Selects imported time base 2 indicated by SRV2 bit field of EQADC_STACCCR register.</td></tr> <tr> <td>11</td><td>Reserved</td></tr> </table>                                                                                                                                                                                                                                | 00 | Selects internally generated time base as time stamp                                         | 01 | Selects imported time base 1 indicated by SRV1 bit field of EQADC_STACCCR register. | 10 | Selects imported time base 2 indicated by SRV2 bit field of EQADC_STACCCR register. | 11 | Reserved |
| 00            | Selects internally generated time base as time stamp                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |    |                                                                                              |    |                                                                                     |    |                                                                                     |    |          |
| 01            | Selects imported time base 1 indicated by SRV1 bit field of EQADC_STACCCR register.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |    |                                                                                              |    |                                                                                     |    |                                                                                     |    |          |
| 10            | Selects imported time base 2 indicated by SRV2 bit field of EQADC_STACCCR register.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |    |                                                                                              |    |                                                                                     |    |                                                                                     |    |          |
| 11            | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |    |                                                                                              |    |                                                                                     |    |                                                                                     |    |          |
| PRE_GAIN[0:1] | <p>ADC Pre-gain control</p> <p>The PRE_GAIN[0:1] controls the gain of the ADC input stage by changing the internal ADC iterations in the gain stage.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |    |                                                                                              |    |                                                                                     |    |                                                                                     |    |          |

*Table continues on the next page...*



**Table 29-119. ADC\_ACR1-14 field descriptions (continued)**

| Field | Description |
|-------|-------------|
| 00    | X1 gain     |
| 01    | X2 gain     |
| 10    | X4 gain     |
| 11    | Reserved    |

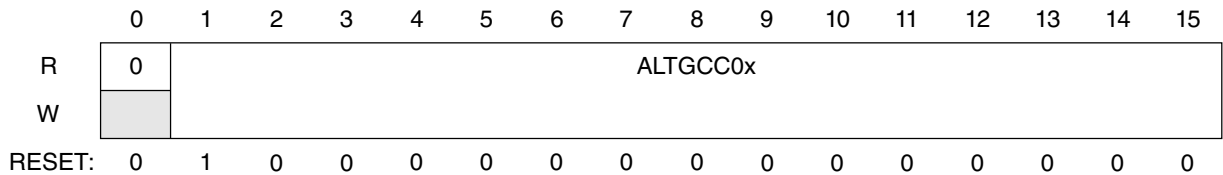
**Table 29-120. Conversion destination selection**

| DAM | DEST[0:3]   | First Destination | Second Destination                 | Description                                                                                                                                                                                                                                                                                                                                            |
|-----|-------------|-------------------|------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0   | 0000        | RFIFOn            | -                                  | The conversion result is sent to the RFIFOn (where n = MESSAGE_TAG).<br><br>The data format is specified by the FFMT bit in the Conversion Command Message.                                                                                                                                                                                            |
|     | 0001 - 0111 | 1-7               | -                                  | The conversion result is sent to the Parallel Side Interface. Up to seven devices can be connected to the parallel interface, using the DEST field to select which one receives the conversion result.<br><br>The data format is specified by the FMTA bit in the Alternate Configuration Control Register.                                            |
|     | 1000        | 1                 | 5                                  | The conversion result is sent to the Parallel Side Interface to two destinations.<br><br>The transferred data is the same for both destinations. <ul style="list-style-type: none"><li>The data format is specified by the FMTA bit in the Alternate Configuration Control Register.</li><li>The FLUSH command is sent to both destinations.</li></ul> |
|     | 1001        | 2                 |                                    |                                                                                                                                                                                                                                                                                                                                                        |
|     | 1010        | 3                 |                                    |                                                                                                                                                                                                                                                                                                                                                        |
|     | 1011        | 4                 |                                    |                                                                                                                                                                                                                                                                                                                                                        |
|     | 1100        | 1                 | 7                                  |                                                                                                                                                                                                                                                                                                                                                        |
|     | 1101        | 2                 |                                    |                                                                                                                                                                                                                                                                                                                                                        |
|     | 1110        | 3                 |                                    |                                                                                                                                                                                                                                                                                                                                                        |
|     | 1111        | 4                 |                                    |                                                                                                                                                                                                                                                                                                                                                        |
| 1   | 0000        | RFIFOn            | Depends on DEST2<br>(Table 29-124) | For DEST = 0, the conversion result is sent to the RFIFOs. The data format is specified by the FFMT bit in the Conversion Command Message.<br><br><b>Note:</b> It is not possible to have FLUSH control for DEST2 >0 because FFMT is used for format of DEST, not for FLUSH.                                                                           |
|     | 0001 - 1111 | 1-15              |                                    | The conversion result is sent to the Parallel Side Interface. Up to fifteen devices can be connected to the parallel interface, using the DEST field to select which one receives the conversion result.<br><br>The data format is specified by the FMTA bit in the Alternate Configuration Control Register.                                          |

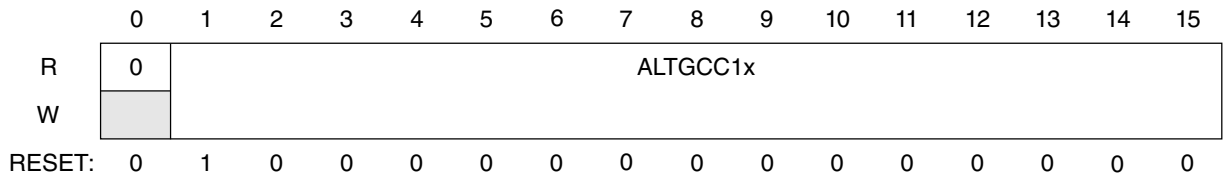
29.5.33.7 ADC0/1 Alternate Gain Registers (ADC0\_AGR1-2 and ADC1\_AGR1-2)

The Alternate Gain Registers (ADC0\_AGRx and ADC1\_AGRx, x=1-2) contain the gain calibration constants used to fine-tune the ADCs conversion results for alternate configurations 1 or 2. A conversion from an ADC uses the corresponding ADC0\_AGRx or ADC1\_AGRx register when the conversion command (with the alternate configuration format) is written to an address in the range 0x08-0x09 of the on-chip ADC memory map. See [ADC calibration feature](#) for details about the calibration scheme used in the EQADC.

ADC0 Register address: 0x31  
ADC0 Register address: 0x35



ADC1 Register address: 0x31  
ADC1 Register address: 0x35



 = Unimplemented or Reserved

Figure 29-110. ADC0/1 Alternate x Gain Register (ADC0/1\_AGRx, x=1-2)

Table 29-121. ADC0/1\_AGRx field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ALTGCC0/1x[0:14] | Alternate Gain Calibration Constant<br><br>ALTGCC0/1x[0:14] contain the gain calibration constants used to fine-tune ADC0/1 conversion results for alternate configurations 1 and 2. The gain calibration constants are 15-bit unsigned fixed point numbers expressed in the <i>GCC_INT.GCC_FRAC</i> binary format. The integer part of the gain constants ( <i>GCC_INT</i> ) contain a single binary digit while their fractional part ( <i>GCC_FRAC</i> ) contain fourteen digits. For details about the GCC data format, see <a href="#">MAC unit and operand data format</a> . |

### 29.5.33.8 ADC0/1 Alternate Offset Register (ADC0\_AOR1-2 and ADC1\_AOR1-2)

The Alternate Offset Registers (ADC0\_AORx and ADC1\_AORx, x=1-2) contain the offset calibration constants used to fine-tune ADCs conversion results for alternate configurations 1 and 2. The offset constants are signed 14-bit integer values. See [ADC calibration feature](#) for details about the calibration scheme used in the EQADC.

ADC0 Register address: 0x32

ADC0 Register address: 0x36



ADC1 Register address: 0x32

ADC1 Register address: 0x36

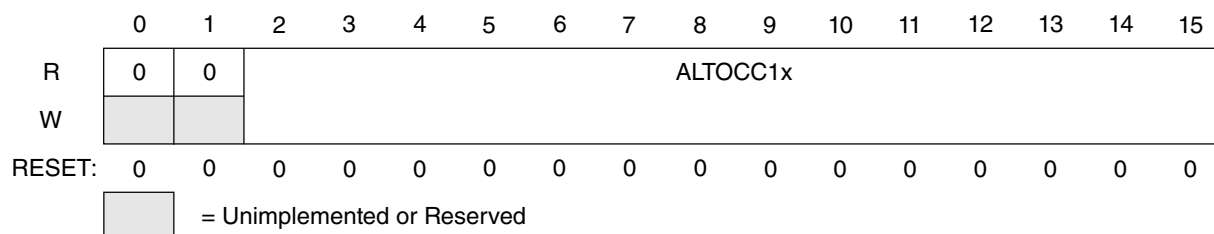


**Figure 29-111. ADC0/1 Alternate x Offset Registers (ADC0/1\_AORx, x=1-2)**

**Table 29-122. ADC0/1\_AORx field descriptions**

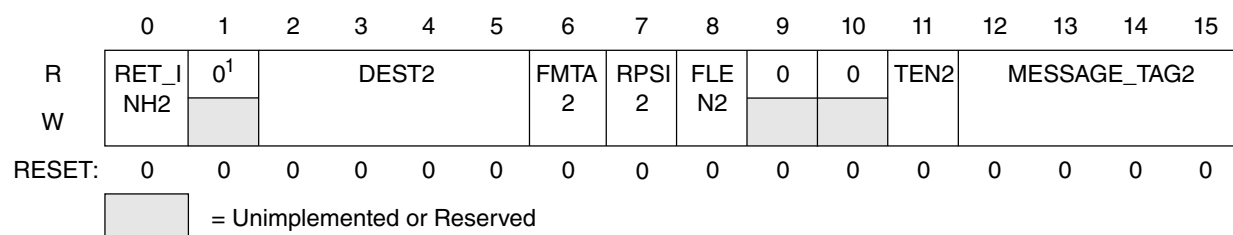
| Field            | Description                                                                                                                                                                                                                                                          |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ALTOCC0/1x[0:13] | Alternate Offset Calibration Constant<br><br>ALTOCC0/1x[0:13] contain the offset calibration constants used to fine-tune ADCs conversion results for alternate configurations 1 or 2. Negative values should be expressed using the two's complement representation. |

### 29.5.33.9 Extended Alternate Configuration 1-14 Control Registers (ADC\_EACR1-14)

The Extended Alternate Configuration Control Registers (ADC\_EACR1-14) are used together with the Alternate Configuration Control Registers (ADC\_ACR1-14) of the same number to complement the alternate configurations of the ADC. A conversion uses one pair of the alternate configuration regs (normal and extended) when the conversion

command (with the alternate conversion format) is written to an address in the range 0x08-0x15 of the on-chip ADC memory map. See [Conversion command format for alternate configurations](#).

ADC0/1 Register address: 0x33  
 ADC0/1 Register address: 0x37  
 ADC0/1 Register address: 0x3B  
 ADC0/1 Register address: 0x3F  
 ADC0/1 Register address: 0x43  
 ADC0/1 Register address: 0x47  
 ADC0/1 Register address: 0x4B  
 ADC0/1 Register address: 0x4F  
 ADC0/1 Register address: 0x53  
 ADC0/1 Register address: 0x57  
 ADC0/1 Register address: 0x5B  
 ADC0/1 Register address: 0x5F  
 ADC0/1 Register address: 0x63  
 ADC0/1 Register address: 0x67



<sup>1</sup> Reserved for expansion of DEST2.

**Figure 29-112. Extended Alternate Configuration 1-14 Control Registers (ADC\_EACR1-14)**

**Table 29-123. ADC\_EACR1-14 field descriptions**

| Field      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RET_INH2   | Result Transfer Inhibit 2 / Decimation Filter Pre-Fill 2<br>Same definition of DEST field but to the peripheral module indicated by DEST2.<br>0 Result transfer to result queue / Decimation Filter in filtering mode<br>1 No result transfer to result queue / Decimation Filter PRE-FILL mode                                                                                                                                                                                                                                                                                                           |
| DEST2[0:3] | Conversion Result Destination 2 Selection<br>The DEST2[0:3] field selects the second destination of the conversion result generated by the Alternate Conversion Command. However this selection only can occur when DAM = 1 (see <a href="#">EQADC Module Configuration Register (EQADC_MCR)</a> ), as shown in previous <a href="#">Table 29-120</a> for DEST field. In addition, the eQADC operation due to this field is also affected by the ADC_ACRx DEST field, by the ADC_EACRx TEN2 field, by MESSAGE_TAG field, by the ADC_EACRx MESSAGE_TAG2 field as defined in <a href="#">Table 29-124</a> . |
| FMTA2      | Conversion Data Format for Alternate Configuration DEST2<br>The FMTA2 bit specifies how the 12-bit conversion data returned by the ADCs is formatted into the 16-bit data which is sent to the RFIFOs or the parallel side interface defined by DEST2. See <a href="#">Table 29-124</a> for more details.<br>0 Right justified unsigned                                                                                                                                                                                                                                                                   |

Table continues on the next page...

**Table 29-123. ADC\_EACR1-14 field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | 1 Right justified signed                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| RPSI2             | <p>Return Parallel Side Interface for DEST2</p> <p>The RPSI2 bit is used in the companion module that has two PSI interfaces (PSI0 and PSI1) that are connected to two eQADCs. It specifies in the companion module if the data received from one PSI should be transmitted to the same PSI or to the other one.</p> <p><b>Note:</b> The companion module may take samples from two eQADCs, allowing double sample rate. If the RPSI bits are different, the results are sent to just one of the eQADCs.</p> <p><b>Note:</b> It is not allowed to use RPSI2=1 when the SoC has only one eQADC. It can occur errors in the companion module due to routing error.</p> <p>0 Companion module results from this eQADC are sent back to the same eQADC using the same PSI.</p> <p>1 Companion module results from this eQADC are sent to the other PSI</p> |
| FLEN2             | <p>Flush Enable to DEST2</p> <p>The FLEN2 bit enables the transfer of the flush command given by the FFMT field from the alternate conversion command when DEST&gt;0.</p> <p>0 No flush command is transmitted to DEST2.</p> <p>1 Flush command is sent to DEST2 if specified by FFMT when DEST&gt;0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| TEN2              | <p>Enable MESSAGE_TAG2 to DEST2</p> <p>This bit specifies if the MESSAGE_TAG2 value should be used to the result transfer of DEST2. See <a href="#">Table 29-124</a> for more details.</p> <p>0 MESSAGE_TAG2 is disabled.</p> <p>1 MESSAGE_TAG2 is enabled.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| MESSAGE_TAG2[0:3] | <p>MESSAGE TAG 2 Field</p> <p>Same as the MESSAGE_TAG field in the Conversion Command as described in <a href="#">Conversion command format for the standard configuration</a>. <a href="#">Table 29-128</a> describes the meaning of the MESSAGE_TAG2.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

**Table 29-124. Conversion Destinations Selection (DAM = 1)**

| DEST<br>[0:3] | DEST2<br>[0:3] | TEN<br>2 | MESSAGE<br>_TAG2 =<br>MESSAGE<br>_TAG | (FMTA2=FFMT)<br>or<br>(FMTA2=FMTA)<br>& (RET2=RET)<br>& (RPSI2=RPSI)<br>&<br>(Flush2=Flush) | First<br>Dest. | Second<br>Dest. | Description                                                                                                                                                                                                  |
|---------------|----------------|----------|---------------------------------------|---------------------------------------------------------------------------------------------|----------------|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0000          | 0000           | 0        | X                                     | X                                                                                           | RFIFOn         | -               | <ul style="list-style-type: none"> <li>The conversion result is sent to the RFIFOn (where n = MESSAGE_TAG), and the data format is specified by the FFMT bit in the alternate conversion command.</li> </ul> |
|               |                | 1        | 1                                     | 1                                                                                           |                |                 |                                                                                                                                                                                                              |

Table continues on the next page...

**Table 29-124. Conversion Destinations Selection (DAM = 1) (continued)**

| DEST<br>[0:3] | DEST2<br>[0:3] | TEN<br>2 | MESSAGE<br>_TAG2 =<br>MESSAGE<br>_TAG | (FMTA2=FFMT)<br>or<br>(FMTA2=FMTA)<br>& (RET2=RET)<br>& (RPSI2=RPSI)<br>&<br>(Flush2=Flush) | First<br>Dest. | Second<br>Dest. | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|---------------|----------------|----------|---------------------------------------|---------------------------------------------------------------------------------------------|----------------|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               |                |          |                                       | 0                                                                                           |                | RFIFOn          | <ul style="list-style-type: none"> <li>Same as above but the conversion result is sent again to RFIFOn if FMTA2 is different from FFMT.</li> <li>OBS: Two values are sent to the same RFIFO for one ADC conversion.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|               |                |          | 0                                     | X                                                                                           |                | RFIFOy          | <ul style="list-style-type: none"> <li>The conversion result is sent to the RFIFOn (where n = MESSAGE_TAG), and the data format is specified by the FFMT bit in the alternate conversion command.</li> <li>The conversion result is also sent to RFIFOy (where y = MESSAGE_TAG2) and the data format is specified by the FMTA2.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                |
| 0000          | 0001 - 1111    | 0        | X                                     | X                                                                                           | RFIFOn         | 1 - 15          | <ul style="list-style-type: none"> <li>The conversion result is sent to the RFIFOn (where n = MESSAGE_TAG), and the data format is specified by the FFMT bit.</li> <li>The conversion result is also sent to PSI using the DEST2 field and the MESSAGE_TAG value. The data format is specified by the FMTA2 bit. There is not flush command. RET2 and RPSI2 are also used.</li> <li>OBS: If the companion block addressed by DEST2 sends data back to the same eQADC, this result is sent to RFIFOn (two values to the same RFIFO for one ADC conversion).</li> </ul> <p><b>Note:</b> It is not possible to have FLUSH control for DEST2&gt;0 because FFMT is used for format of DEST, not for FLUSH.</p> |
|               |                | 1        | 1                                     | X                                                                                           |                |                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|               |                |          | 0                                     | X                                                                                           |                |                 | <ul style="list-style-type: none"> <li>Same as above, but if the companion block addressed by DEST2 sends data back to the eQADC, this data is sent to RFIFOy given by MESSAGE_TAG2.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 0001 - 1111   | 0000           | 0        | X                                     | X                                                                                           | 1 - 15         | -               | <ul style="list-style-type: none"> <li>The conversion result is sent to PSI using the DEST field and the MESSAGE_TAG value. The data</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

Table continues on the next page...

**Table 29-124. Conversion Destinations Selection (DAM = 1) (continued)**

| DEST<br>[0:3]  | DEST2<br>[0:3]                              | TEN<br>2 | MESSAGE<br>_TAG2 =<br>MESSAGE<br>_TAG | (FMTA2=FFMT)<br>or<br>(FMTA2=FMTA)<br>& (RET2=RET)<br>& (RPSI2=RPSI)<br>&<br>(Flush2=Flush) | First<br>Dest. | Second<br>Dest.                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|----------------|---------------------------------------------|----------|---------------------------------------|---------------------------------------------------------------------------------------------|----------------|----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                |                                             |          |                                       |                                                                                             |                |                                  | format is specified by the FMTA bit. The flush command depends on FFMT command field.                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|                |                                             | 1        | 1                                     | X                                                                                           |                | RFIFOn                           | <ul style="list-style-type: none"> <li>The conversion result is sent to PSI using the DEST field and the MESSAGE_TAG value. The data format is specified by the FMTA bit. The flush command depends on FFMT command field.</li> <li>The conversion result is also sent to the RFIFOn (where n = MESSAGE_TAG2), and the data format is specified by the FMTA2 bit field.</li> <li>OBS: If the companion block addressed by DEST sends data back to the same eQADC, this result is sent to RFIFOn (two values to the same RFIFO for one ADC conversion).</li> </ul> |
|                |                                             |          | 0                                     | X                                                                                           |                | RFIFOy                           | <ul style="list-style-type: none"> <li>Same as above, but if the companion block addressed by DEST sends data back to the eQADC, this data is sent to RFIFOn=MESSAGE_TAG.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                              |
|                |                                             |          |                                       |                                                                                             |                |                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 0001 -<br>1111 | 0001 -<br>1111<br><br>(Equal<br>to<br>DEST) | 0        | X                                     | 1                                                                                           | 1 - 15         | -                                | <ul style="list-style-type: none"> <li>The conversion result is sent to PSI using the DEST field and the MESSAGE_TAG value. The data format is specified by the FMTA bit. The flush command depends on FFMT command field.</li> </ul>                                                                                                                                                                                                                                                                                                                             |
|                |                                             |          |                                       | 0                                                                                           |                |                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|                |                                             | 1        | 1                                     | 0                                                                                           |                | 1 - 15<br>(Equal to<br>1st Dest) | <ul style="list-style-type: none"> <li>Same as above.</li> <li>The conversion result is sent again to PSI to the same DEST=DEST2. However, the PSI data bus value is different.</li> <li>OBS: The data related to DEST is sent first. And the data related to DEST2 is sent later.</li> </ul>                                                                                                                                                                                                                                                                     |
|                |                                             |          |                                       | 1                                                                                           |                | -                                | <ul style="list-style-type: none"> <li>The conversion result is sent to PSI using the DEST field and the MESSAGE_TAG value. The data</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                   |

Table continues on the next page...

**Table 29-124. Conversion Destinations Selection (DAM = 1) (continued)**

| DEST<br>[0:3]  | DEST2<br>[0:3]                              | TEN<br>2 | MESSAGE<br>_TAG2 =<br>MESSAGE<br>_TAG | (FMTA2=FFMT)<br>or<br>(FMTA2=FMTA)<br>& (RET2=RET)<br>& (RPSI2=RPSI)<br>&<br>(Flush2=Flush) | First<br>Dest. | Second<br>Dest.                         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|----------------|---------------------------------------------|----------|---------------------------------------|---------------------------------------------------------------------------------------------|----------------|-----------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                |                                             |          |                                       |                                                                                             |                |                                         | format is specified by the FMTA bit.<br>The flush command depends on<br>FFMT command field.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|                |                                             |          | 0                                     | X                                                                                           |                | 1 - 15<br>(Equal to<br>1st Dest)        | <ul style="list-style-type: none"> <li>Same as above.</li> <li>The conversion result is also sent to PSI using the DEST2=DEST field and with the MESSAGE_TAG2 value. The data format is specified by the FMTA2 bit. The flush command depends on FLEN2 bit field. RET2 and RPSI2 are also used.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 0001 -<br>1111 | 0001 -<br>1111<br>(Not<br>equal to<br>DEST) | 0        | X                                     | 0                                                                                           | 1-15           | 1 - 15<br>(Not<br>equal to<br>1st Dest) | <ul style="list-style-type: none"> <li>The conversion result is sent to PSI using the DEST field and the MESSAGE_TAG value. The data format is specified by the FMTA bit. The flush command depends on FFMT command field.</li> <li>The conversion result is also sent to PSI using the DEST2 (not equal DEST) field and with the MESSAGE_TAG2 value. The data format is specified by the FMTA2 bit. The flush command depends on FLEN2 bit field. RET2 and RPSI2 are also used.</li> <li>OBS1: If the companion blocks addressed by DEST and DEST2 send data back to the same eQADC, these results are sent to the same RFIFO (two values to the same RFIFO for one ADC conversion).</li> <li>OBS2: The data related to DEST is sent first. And the data related to DEST2 is sent later.</li> </ul> <p><b>Note:</b> When sending timestamp in this case, the values sent to RFIFO can be out of order due to the prioritization of reading in the receiving part of the PSI sub-block.</p> |
|                |                                             |          |                                       | 1                                                                                           |                |                                         | <ul style="list-style-type: none"> <li>Same as above, except OBS2.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|                |                                             | 1        | 1                                     | 1                                                                                           |                |                                         | In this case, the data is sent to DEST and DEST2 at the same time.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

Table continues on the next page...



**Table 29-124. Conversion Destinations Selection (DAM = 1) (continued)**

| DEST<br>[0:3] | DEST2<br>[0:3] | TEN<br>2 | MESSAGE<br>_TAG2 =<br>MESSAGE<br>_TAG | (FMTA2=FFMT)<br>or<br>(FMTA2=FMTA)<br>& (RET2=RET)<br>& (RPSI2=RPSI)<br>&<br>(Flush2=Flush) | First<br>Dest. | Second<br>Dest. | Description                                                                                  |
|---------------|----------------|----------|---------------------------------------|---------------------------------------------------------------------------------------------|----------------|-----------------|----------------------------------------------------------------------------------------------|
|               |                |          |                                       | 0                                                                                           |                |                 | <ul style="list-style-type: none"> <li>Same as first case above.</li> </ul>                  |
|               |                |          | 0                                     | X                                                                                           |                |                 | <ul style="list-style-type: none"> <li>Same as first case, except OBS1 and Note2.</li> </ul> |

### 29.5.33.10 ADC Pull Up/Down Control Register x (ADC\_PUDCRx, x=0-39)

The ADC Pull Up/Down Control Register  $x$  (ADC\_PUDCR $x$ ) contains configuration bits for pull up and pull down resistors present at ADC input channels  $x$ ,  $x=0$  to 39.

ADC0/1 Register address: 0x70-0x97

|        |   |   |          |   |   |   |           |   |   |   |    |    |    |    |    |    |
|--------|---|---|----------|---|---|---|-----------|---|---|---|----|----|----|----|----|----|
|        | 0 | 1 | 2        | 3 | 4 | 5 | 6         | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R      | 0 | 0 | CH_PULLx |   | 0 | 0 | PULL_STRx |   | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |
| W      |   |   |          |   |   |   |           |   |   |   |    |    |    |    |    |    |
| RESET: | 0 | 0 | 0        | 0 | 0 | 0 | 0         | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

 = Unimplemented or Reserved

**Figure 29-113. ADC Pull Up/Down Control Register x (ADC\_PUDCRx, x=0-39)****Table 29-125. ADC\_PUDCRx field descriptions**

| Field          | Description                                                                                                                                                                                                                                                                                                                                                           |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CH_PULLx[0:1]  | <p>Channel x Pull Up/Down Control bits</p> <p>The CH_PULLx[0:1] field controls the pull up/down configuration of the channel x.</p> <p>00 No Pull resistors connected to the channel</p> <p>01 Pull Up resistor connected to the channel</p> <p>10 Pull Down resistor connected to the channel</p> <p>11 Pull Up and Pull Down resistors connected to the channel</p> |
| PULL_STRx[0:1] | <p>Pull Up/Down Strength Control bits of channel x</p> <p>The PULL_STRx[0:1] bit field defines the strength of the channel x pull up or down resistors.</p> <p>00 Reserved</p> <p>01 200 Kohms pull resistor</p>                                                                                                                                                      |

*Table continues on the next page...*

Table 29-125. ADC\_PUDCRx field descriptions (continued)

| Field | Description                                                                    |
|-------|--------------------------------------------------------------------------------|
| 10    | 100 Kohms pull resistor                                                        |
| 11    | 5 Kohms (Approx.) pull resistor (this set is not available for CH_PULL_x = 11) |

29.5.33.11 STAC Bus Time Base 1/2 Registers (ADC\_STACTBR1/2)

The STAC Bus Time Base Register1/2 (ADC\_STACTBR1/2) contain the current value of the imported time base counter as selected by bit fields on register EQADC\_STACCCR. See [n Conversion command format for the standard configuration](#) for more details.

ADC0/1 Register address: 0xA0, 0xA1

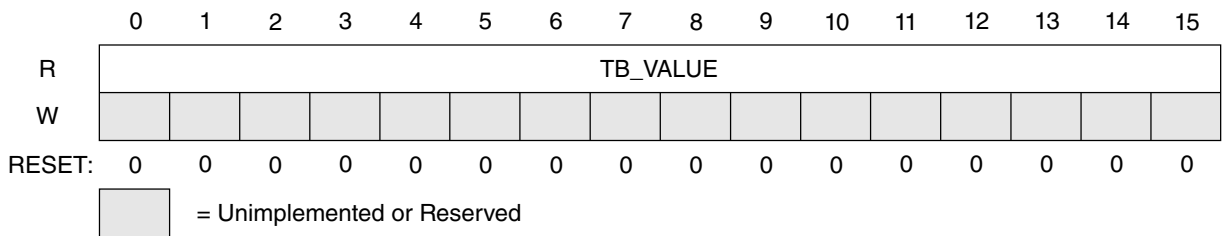


Figure 29-114. ADC Time Base Counter Register (ADC\_TBCR)

Table 29-126. ADC\_TBCR field descriptions

| Field          | Description                                                                                       |
|----------------|---------------------------------------------------------------------------------------------------|
| TB_VALUE[0:15] | Time Base VALUE Field<br>The TB_VALUE field contains the current value of the imported time base. |

29.6 Functional description

29.6.1 Overview

The EQADC provides a parallel interface to two on-chip ADCs and a parallel side interface to on-chip companion modules, like a decimation filter. The two on-chip ADCs are architected to allow access to all the analog channels.

Initially, command data is contained in system memory in a user defined data structure which is likely to be a queue as depicted in [Figure 29-1<sup>3</sup>](#). Command data is moved between the CQueues and CFIFOs by the host CPU or by the DMAC which respond to interrupt and DMA requests generated by the EQADC. The EQADC supports software and hardware triggers from other blocks or external pins to initiate transfers of commands from the multiple CFIFOs to the on-chip ADCs.

CFIFOs can be configured in single-scan or continuous-scan mode. When a CFIFO is configured in single-scan mode, the EQADC scans the CQueue one time. The EQADC stops transferring commands from the triggered CFIFO after detecting the EOQ bit set in the last transfer. After an EOQ bit is detected, software involvement is required to rearm the CFIFO so that it can detect new trigger events.

When a CFIFO is configured for continuous-scan mode, the whole CQueue is scanned multiple times. After the detection of an asserted EOQ bit in the last command transfer, command transfers can continue or not depending on the mode of operation of the CFIFO.

CFIFO0 has a special configuration option to allow a repetitive sequence of conversion commands (streaming mode) with high priority characteristics (abort operation) or not. This feature is useful with the immediate conversion command feature that allows the immediate execution of a conversion command or a sequence of commands with critical timing even with the possibility of abortion of some current ADC conversion in progress. The aborted command is stored and executed again as soon as the critical timing commands have been finished.

The multiple Result FIFOs (RFIFOs) can receive data from the on-chip ADCs or from on-chip companion modules. Data from the on-chip ADCs can be routed to the side interface, processed by the on-chip companion modules and then routed back through the side interface to the RFIFOs.

## 29.6.2 Data flow in EQADC

### 29.6.2.1 Overview and basic terminology

[Figure 29-115](#) shows how command data flows inside the EQADC system. A Command Message is the predefined format at which command data is stored on the CQueues. A Command message has 32 bits and is composed of two parts: a CFIFO header and an ADC Command. Command messages are moved from the CQueues to the CFIFOs by the

3. Command and result data can be stored in system memory in any user defined data structure. However, in this document it will be assumed that the data structure of choice is a queue, since it is the most likely data structure to be used and because queues are the only type of data structure supported by the DMAC.

host CPU or by the DMAC as they respond to interrupt and DMA requests generated by the EQADC. The EQADC generates these requests whenever a CFIFO is not full. *The FIFO Control Unit* will only transfer to a CBuffer the ADC command part of the Command Message. Information in the CFIFO header together with the upper bit of the ADC command is used by the *FIFO Control Unit* to arbitrate which triggered CFIFO will be transferring the next command.

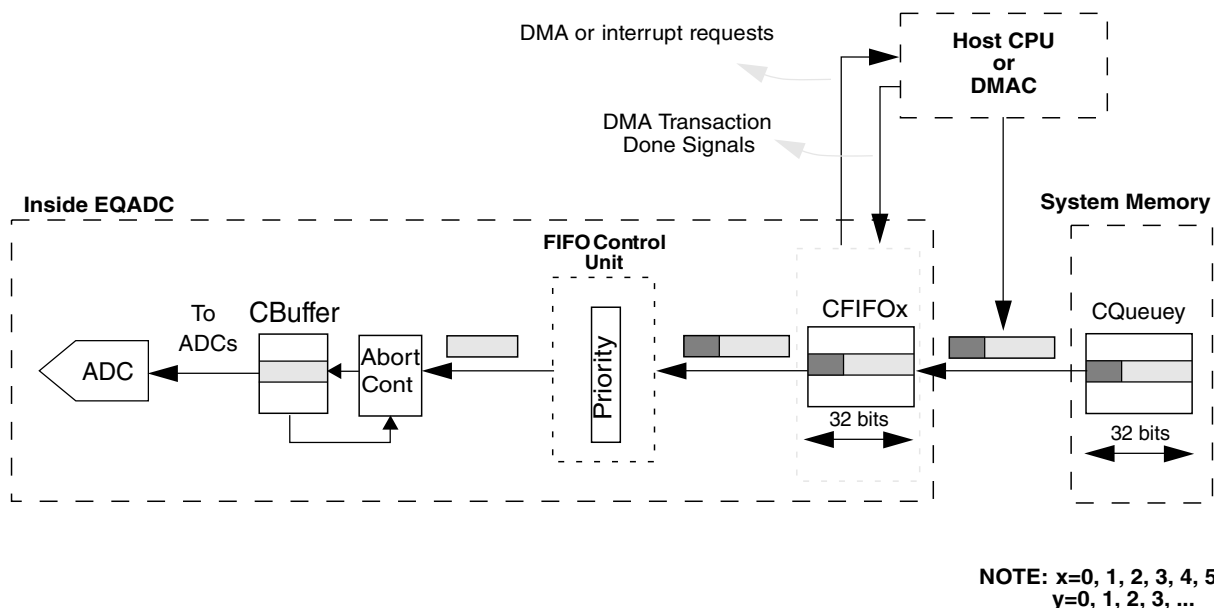


**Figure 29-115. Command Flow during EQADC operation**

ADC commands sent to the on-chip CBuffers are executed in a first-in-first-out basis with exception when the immediate conversion command function is enabled. Three types of results can be expected: data read from an ADC register, a conversion result, or a time stamp.

### Note

While the EQADC pops commands out from a CFIFO, it also is checking the number of entries in the CFIFO and generating requests to fill it. The process of pushing and popping commands to and from a CFIFO can occur simultaneously. However, this is not true for CFIFO0 when configured to operate in streaming mode for popping.

The *FIFO Control Unit* expects all incoming results to be shaped in a predefined Result Message format. [Figure 29-116](#) shows how result data flows inside the EQADC system. Results generated on the on-chip ADCs are adjusted considering the selected resolution of the ADC and are formatted into result messages inside the *Result Format and Calibration Sub-Block*. This result message can be routed directly to one of the RFIFOs or to an on-chip companion module via the parallel side interface. After the data is processed by the companion module, it can be routed back to one of the RFIFOs via the side interface with the correct format. A result message is composed of an RFIFO header and an ADC Result. The *FIFO Control Unit* decodes the information contained in the RFIFO header to determine the RFIFO to which the ADC result should be sent. Once in an RFIFO, the ADC result is moved to the corresponding RQueue by the host CPU or by the DMAC as they respond to interrupt and DMA requests generated by the EQADC. The EQADC generates these requests whenever an RFIFO has at least one entry.

### Note

While conversion results are returned, the EQADC is checking the number of entries in the RFIFO and generating requests to empty it. The process of pushing and popping ADC results to and from an RFIFO can occur simultaneously.

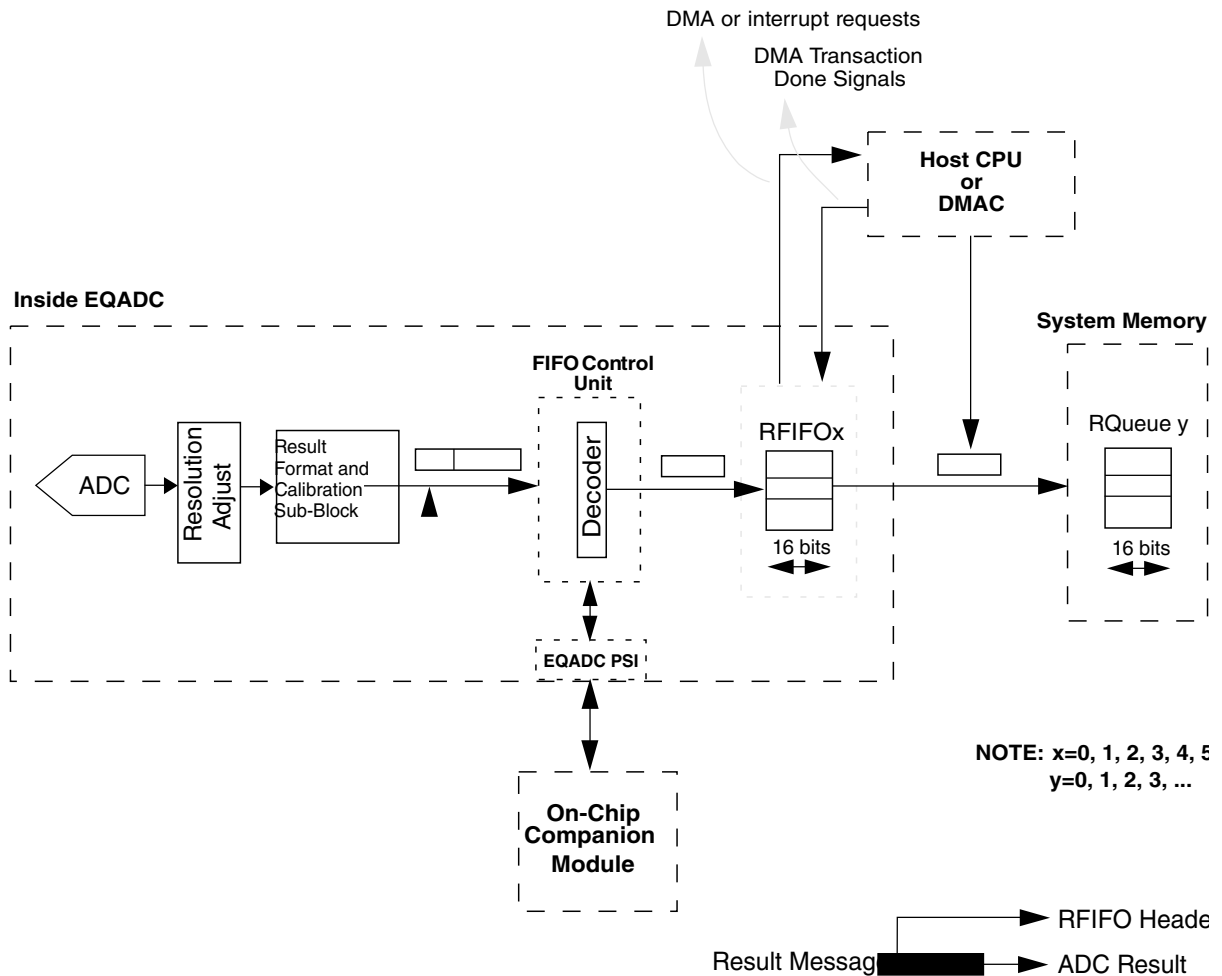


Figure 29-116. Result Flow during EQADC operation

### 29.6.2.2 Message format in EQADC

This section explains the command and result message formats used for on-chip ADC operation

A Command Message is the predefined format at which command data is stored on the CQueues. A Command message has 32 bits and is composed of two parts: a CFIFO header and an ADC Command. The size of the CFIFO header is fixed to 6 bits, and it works as inputs to the *FIFO Control Unit*. It controls when a CQueue ends, when it pauses. Information contained in the CFIFO header, together with the upper bit of the ADC Command is used by the *FIFO Control Unit* to arbitrate which triggered CFIFO will transfer the next command. ADC commands are encoded inside the least significant 26 bits of the command message.

A Result message is composed of an RFIFO header and an ADC Result. The *FIFO Control Unit* decodes the information contained in the RFIFO header to determine the RFIFO to which the ADC result should be sent. An ADC result is always 16 bits long.

### 29.6.2.2.1 Message Formats for On-Chip ADC Operation

This section describes the Command/Result message formats used for on-chip ADC operation.

#### 29.6.2.2.1.1 Conversion command format for the standard configuration

Figure 29-117 describes the format for conversion commands when interfacing with the on-chip ADCs in the standard configuration. The standard configuration is selected when the lowest byte (bits 24-31) of the conversion command is set to zero. In the standard configuration, the conversion result is always routed to one of the RFIFOs. A time stamp information can be optionally requested.

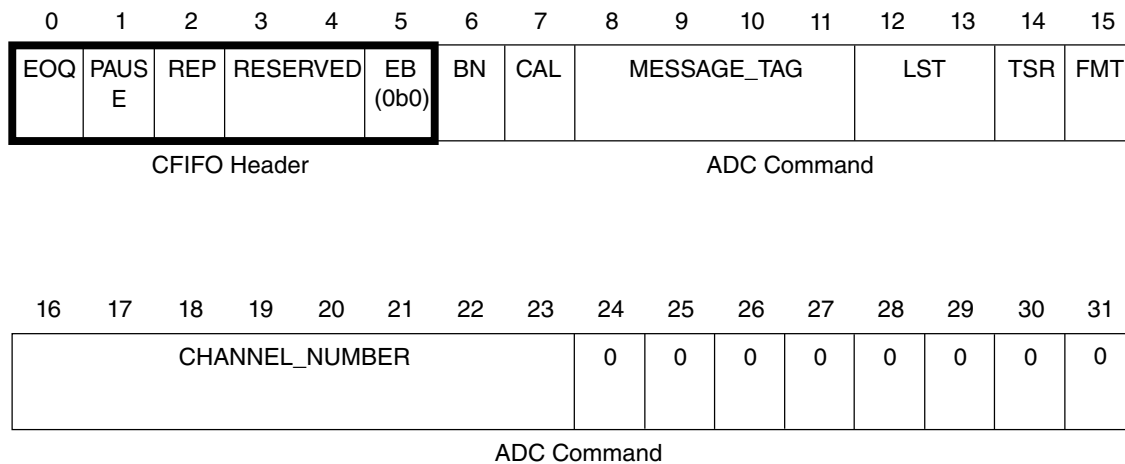


**Figure 29-117. Conversion command format for the standard configuration**

**Table 29-127. Conversion command format for the standard configuration — field descriptions**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| EOQ   | <p>End Of Queue Bit</p> <p><b>NOTE:</b> The EOQ bit is asserted in the last command of a CQueue to indicate to the EQADC that a scan of the CQueue is completed. EOQ instructs the EQADC to reset its current CFIFO transfer counter value (TC_CF) to zero. Depending on the CFIFO mode of operation, the CFIFO status will also change upon the detection of an asserted EOQ bit on the last transferred command - see <a href="#">CFIFO scan trigger modes</a> for details. If both the PAUSE and EOQ bits are asserted in the same command message the respective flags are set, but the CFIFO status changes as if only the EOQ bit were asserted.</p> <p>0 Not the last entry of the CQueue.</p> |

Table continues on the next page...

**Table 29-127. Conversion command format for the standard configuration — field descriptions (continued)**

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                  | 1 Last entry of the CQueue.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| PAUSE            | <p>Pause Bit</p> <p>The Pause bit allows software to create sub-queues within a CQueue. When the EQADC completes the transfer of a command with an asserted Pause bit, the CFIFO enters the WAITING FOR TRIGGER state. See <a href="#">CFIFO operation status</a> for a description of the state transitions. The Pause bit is only valid when CFIFO operation mode is configured to single or continuous-scan edge trigger mode.</p> <p><b>Note:</b> If both the PAUSE and EOQ bits are asserted in the same command message the respective flags are set, but the CFIFO status changes as if only the EOQ bit were asserted.</p> <p>0 Do not enter WAITING FOR TRIGGER state after transfer of the current Command Message.</p> <p>1 Enter WAITING FOR TRIGGER state after transfer of the current Command Message.</p>                                             |
| REP              | <p>Repeat/loop Start Point Indication Bit</p> <p>The REP bit is asserted in the command to indicate where is the start point of the sub-queue to be repeated when the streaming mode is enabled. The PAUSE bit indicates the end point of the sub-queue. Therefore, both can occur in the same command or in separated ones. If two or more REP bits are read before a PAUSE bit, this is an error case and the intermediary REP bits are ignored.</p> <p>0 It is not the start point of a loop.</p> <p>1 Indicates the start point of the sub-queue to be repeated.</p>                                                                                                                                                                                                                                                                                              |
| Reserved<br>BN   | <p>Reserved</p> <p>Buffer Number Bit</p> <p>BN indicates which buffer the message will be stored in.</p> <p>0 Message stored in buffer 0.</p> <p>1 Message stored in buffer 1.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| CAL              | <p>CALibration Bit</p> <p>CAL indicates if the returning conversion result must be calibrated.</p> <p>0 Do not calibrate conversion result.</p> <p>1 Calibrate conversion result.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| MESSAGE_TAG[0:3] | <p>MESSAGE_TAG Field</p> <p>The MESSAGE_TAG allows the EQADC to separate returning results into different RFIFOs. <a href="#">Table 29-128</a> describes the meaning of the MESSAGE_TAG. When the EQADC transfers a command, the MESSAGE_TAG is included as part of the command. Eventually on-chip ADC returns the result with the same MESSAGE_TAG. The EQADC separates incoming messages into different RFIFOs by decoding the MESSAGE_TAG of the incoming data.</p> <p><b>Note:</b> The MESSAGE_TAG value can be particularly used in a companion module (that uses the PSI) as the address to the annexed conversion result. The data can be routed to different sub-blocks based on the MESSAGE_TAG, similar to what is done with the RFIFOs in the eQADC. Additionally, in the companion module it is not required to follow <a href="#">Table 29-128</a>.</p> |
| LST[0:1]         | Long Sampling Time                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

*Table continues on the next page...*



**Table 29-127. Conversion command format for the standard configuration — field descriptions (continued)**

| Field               | Description                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                     | <p>These two bits determine the duration of the sampling time in ADC clock cycles.</p> <p>00          2 sampling cycles (ADC clock cycles)</p> <p>01          8 sampling cycles (ADC clock cycles)</p> <p>10          64 sampling cycles (ADC clock cycles)</p> <p>11          128 sampling cycles (ADC clock cycles)</p>                                                                                                           |
| TSR                 | <p>Time Stamp Request</p> <p>TSR indicates the request for a time stamp. When TSR is asserted, the on-chip <i>ADC Control Logic</i> returns a time stamp for the current conversion command after the conversion result is sent to the RFIFOs. See <a href="#">Time stamp feature</a> for details.</p> <p>0          Return conversion result only.</p> <p>1          Return conversion time stamp after the conversion result.</p> |
| FMT                 | <p>Conversion Data Format</p> <p>FMT specifies to the EQADC how to format the 12-bit conversion data returned by the ADCs into the 16-bit format which is sent to the RFIFOs. See <a href="#">ADC result format for on-chip ADC operation</a> for details.</p> <p>0          Right justified unsigned.</p> <p>1          Right justified signed.</p>                                                                                |
| CHANNEL_NUMBER[0:7] | <p>Channel Number Field</p> <p>The CHANNEL_NUMBER field selects the analog input channel. The software programs this field with the channel number corresponding to the analog input pin to be sampled and converted. See <a href="#">Channel assignment</a> for details.</p>                                                                                                                                                       |

**Table 29-128. MESSAGE\_TAG description**

| MESSAGE_TAG[0:3] | MESSAGE_TAG Meaning                    |
|------------------|----------------------------------------|
| 0b0000           | Result is sent to RFIFO 0              |
| 0b0001           | Result is sent to RFIFO 1              |
| 0b0010           | Result is sent to RFIFO 2              |
| 0b0011           | Result is sent to RFIFO 3              |
| 0b0100           | Result is sent to RFIFO 4              |
| 0b0101           | Result is sent to RFIFO 5              |
| 0b0110 - 0b0111  | Reserved                               |
| 0b1000           | Null Message Received                  |
| 0b1001           | Reserved for customer use <sup>1</sup> |
| 0b1010           | Reserved for customer use <sup>2</sup> |
| 0b1011 - 0b1111  | Reserved                               |

1. These messages are treated as null messages. Therefore, they must obey the format for incoming null messages and return valid BUSY0/1 fields. "

29.6.2.2.1.2 Conversion command format for alternate configurations

Figure 29-118 describes the format for conversion commands when interfacing with the on-chip ADCs in one of the 14 alternate configurations. An alternate configuration is selected when the lowest byte (bits 24-31) of the conversion command is set to a value in the range 0x08-0x15. Each value in this range selects one of the 14 alternate configuration (0x08 selects Alternate Configuration 1, 0x15 selects Alternate Configuration 14). In the alternate configurations, the conversion result can be routed to one of the RFIFOs and/or to the parallel side interface to communicate with an on-chip companion module, or to one or two on-chip companion modules. Bit fields in the corresponding Alternate Configuration Control Register and Extended Alternate Configuration Control Register select the Internal RFIFO or Parallel Side Interface as the destination for the conversion result. A time stamp information can be optionally requested.

All fields, except FFMT and ALT\_CONFIG\_SEL, are identical to the ones in the standard configuration format. Only the fields which are different from the standard format will be described here.

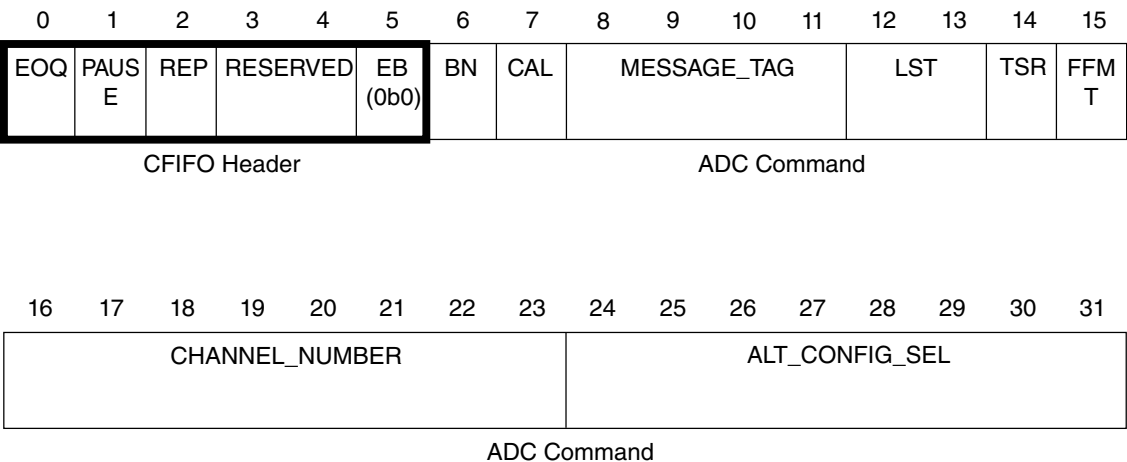


Figure 29-118. Conversion command format for alternate configurations

Table 29-129. Conversion command format for alternate configurations — field descriptions

| Field       | Description      |
|-------------|------------------|
| EOQ         | See Table 29-127 |
| PAUSE       | See Table 29-127 |
| REP         | See Table 29-127 |
| EB          | See Table 29-127 |
| BN          | See Table 29-127 |
| CAL         | See Table 29-127 |
| MESSAGE_TAG | See Table 29-127 |

Table continues on the next page...

**Table 29-129. Conversion command format for alternate configurations — field descriptions (continued)**

| Field               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |      |                                                                                                                                                                       |      |                                                                                                                                                                    |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                            |      |                            |      |                            |      |                            |      |                            |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|---------------------------|------|---------------------------|------|---------------------------|------|---------------------------|------|---------------------------|------|---------------------------|------|---------------------------|------|----------------------------|------|----------------------------|------|----------------------------|------|----------------------------|------|----------------------------|
| LST                 | See <a href="#">Table 29-127</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |      |                                                                                                                                                                       |      |                                                                                                                                                                    |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                            |      |                            |      |                            |      |                            |      |                            |
| TSR                 | See <a href="#">Table 29-127</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |      |                                                                                                                                                                       |      |                                                                                                                                                                    |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                            |      |                            |      |                            |      |                            |      |                            |
| FFMT                | <p>Flush or Format</p> <p>The function of this bit depends on the DEST field of the Alternate Configuration Control Register. If DEST is equal to 0b000, then FFMT defines the format in which the 12-bit conversion result are stored in the RFIFOs. If DEST is not equal to 0b000, then the FFMT bit is used to send a flush (soft-reset) signal through the parallel side interface to the companion module addressed by the DEST field.</p> <p>In case DEST is not equal to 0b000, the FMTA bit in the Alternate Configuration Control register is used to define the conversion result format.</p> <p><b>Note:</b> The flush signal can be asserted along with a valid conversion result. In this case the companion module should execute the software-reset first and then consider the conversion result as a valid data for the filtering algorithm.</p> <table> <tr> <td>0</td><td>Conversion Result Format set to right justified unsigned if DEST is equal to 0b000. No flush signal is sent through the side interface if DEST is not equal to 0b000.</td></tr> <tr> <td>1</td><td>Conversion Result Format set to right justified signed if DEST is equal to 0b000. A flush signal is sent through the side interface if DEST is not equal to 0b000.</td></tr> </table> | 0    | Conversion Result Format set to right justified unsigned if DEST is equal to 0b000. No flush signal is sent through the side interface if DEST is not equal to 0b000. | 1    | Conversion Result Format set to right justified signed if DEST is equal to 0b000. A flush signal is sent through the side interface if DEST is not equal to 0b000. |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                            |      |                            |      |                            |      |                            |      |                            |
| 0                   | Conversion Result Format set to right justified unsigned if DEST is equal to 0b000. No flush signal is sent through the side interface if DEST is not equal to 0b000.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |      |                                                                                                                                                                       |      |                                                                                                                                                                    |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                            |      |                            |      |                            |      |                            |      |                            |
| 1                   | Conversion Result Format set to right justified signed if DEST is equal to 0b000. A flush signal is sent through the side interface if DEST is not equal to 0b000.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |      |                                                                                                                                                                       |      |                                                                                                                                                                    |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                            |      |                            |      |                            |      |                            |      |                            |
| CHANNEL_NUMBER[0:7] | See <a href="#">Table 29-127</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |      |                                                                                                                                                                       |      |                                                                                                                                                                    |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                            |      |                            |      |                            |      |                            |      |                            |
| ALT_CONFIG_SEL[]    | <p>Alternate Configuration Selection</p> <p>This field selects one of the alternate configurations.</p> <table> <tr><td>0x08</td><td>Alternate configuration 1</td></tr> <tr><td>0x09</td><td>Alternate configuration 2</td></tr> <tr><td>0x0A</td><td>Alternate configuration 3</td></tr> <tr><td>0x0B</td><td>Alternate configuration 4</td></tr> <tr><td>0x0C</td><td>Alternate configuration 5</td></tr> <tr><td>0x0D</td><td>Alternate configuration 6</td></tr> <tr><td>0x0E</td><td>Alternate configuration 7</td></tr> <tr><td>0x0F</td><td>Alternate configuration 8</td></tr> <tr><td>0x10</td><td>Alternate configuration 9</td></tr> <tr><td>0x11</td><td>Alternate configuration 10</td></tr> <tr><td>0x12</td><td>Alternate configuration 11</td></tr> <tr><td>0x13</td><td>Alternate configuration 12</td></tr> <tr><td>0x14</td><td>Alternate configuration 13</td></tr> <tr><td>0x15</td><td>Alternate configuration 14</td></tr> </table>                                                                                                                                                                                                                                                                                                                           | 0x08 | Alternate configuration 1                                                                                                                                             | 0x09 | Alternate configuration 2                                                                                                                                          | 0x0A | Alternate configuration 3 | 0x0B | Alternate configuration 4 | 0x0C | Alternate configuration 5 | 0x0D | Alternate configuration 6 | 0x0E | Alternate configuration 7 | 0x0F | Alternate configuration 8 | 0x10 | Alternate configuration 9 | 0x11 | Alternate configuration 10 | 0x12 | Alternate configuration 11 | 0x13 | Alternate configuration 12 | 0x14 | Alternate configuration 13 | 0x15 | Alternate configuration 14 |
| 0x08                | Alternate configuration 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |      |                                                                                                                                                                       |      |                                                                                                                                                                    |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                            |      |                            |      |                            |      |                            |      |                            |
| 0x09                | Alternate configuration 2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |      |                                                                                                                                                                       |      |                                                                                                                                                                    |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                            |      |                            |      |                            |      |                            |      |                            |
| 0x0A                | Alternate configuration 3                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |      |                                                                                                                                                                       |      |                                                                                                                                                                    |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                            |      |                            |      |                            |      |                            |      |                            |
| 0x0B                | Alternate configuration 4                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |      |                                                                                                                                                                       |      |                                                                                                                                                                    |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                            |      |                            |      |                            |      |                            |      |                            |
| 0x0C                | Alternate configuration 5                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |      |                                                                                                                                                                       |      |                                                                                                                                                                    |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                            |      |                            |      |                            |      |                            |      |                            |
| 0x0D                | Alternate configuration 6                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |      |                                                                                                                                                                       |      |                                                                                                                                                                    |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                            |      |                            |      |                            |      |                            |      |                            |
| 0x0E                | Alternate configuration 7                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |      |                                                                                                                                                                       |      |                                                                                                                                                                    |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                            |      |                            |      |                            |      |                            |      |                            |
| 0x0F                | Alternate configuration 8                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |      |                                                                                                                                                                       |      |                                                                                                                                                                    |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                            |      |                            |      |                            |      |                            |      |                            |
| 0x10                | Alternate configuration 9                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |      |                                                                                                                                                                       |      |                                                                                                                                                                    |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                            |      |                            |      |                            |      |                            |      |                            |
| 0x11                | Alternate configuration 10                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |      |                                                                                                                                                                       |      |                                                                                                                                                                    |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                            |      |                            |      |                            |      |                            |      |                            |
| 0x12                | Alternate configuration 11                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |      |                                                                                                                                                                       |      |                                                                                                                                                                    |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                            |      |                            |      |                            |      |                            |      |                            |
| 0x13                | Alternate configuration 12                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |      |                                                                                                                                                                       |      |                                                                                                                                                                    |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                            |      |                            |      |                            |      |                            |      |                            |
| 0x14                | Alternate configuration 13                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |      |                                                                                                                                                                       |      |                                                                                                                                                                    |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                            |      |                            |      |                            |      |                            |      |                            |
| 0x15                | Alternate configuration 14                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |      |                                                                                                                                                                       |      |                                                                                                                                                                    |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                           |      |                            |      |                            |      |                            |      |                            |      |                            |

### 29.6.2.2.1.3 Write configuration command format for on-chip ADC operation

Figure 29-119 describes the command message format for a write configuration command when interfacing with the on-chip ADCs. A write configuration command is used to set the control registers of the on-chip ADCs. No conversion data will be returned for a write configuration command. Write configuration commands are differentiated from read configuration commands by a negated R/W bit.

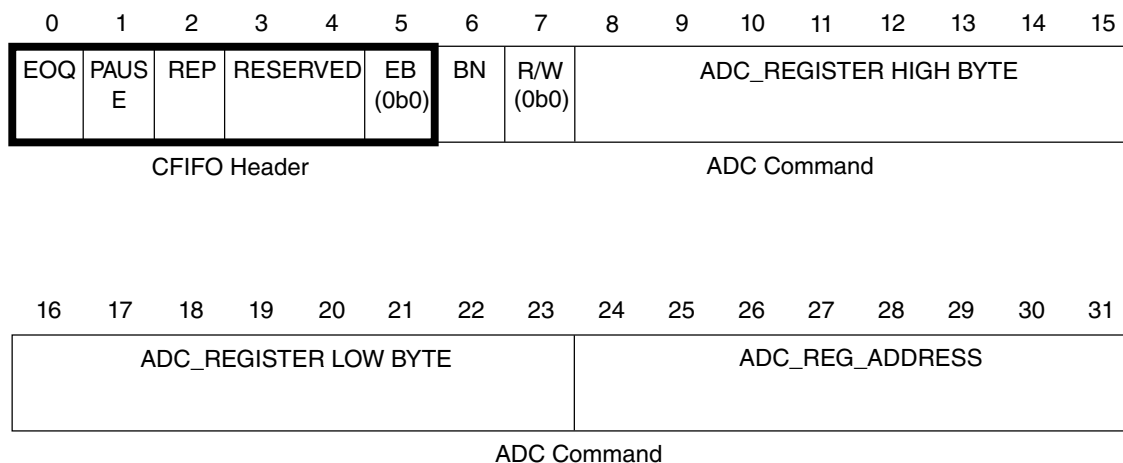


**Figure 29-119. Write configuration command format for on-chip ADC operation**

**Table 29-130. Write configuration command format for on-chip ADC operation — field descriptions**

| Field                       | Description                                                                                                                                                                    |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| EOQ                         | End Of Queue Bit                                                                                                                                                               |
| PAUSE                       | Pause Bit                                                                                                                                                                      |
| REP                         | Repeat/loop Start Point Indication Bit                                                                                                                                         |
| BN                          | Buffer Number Bit<br>See <a href="#">Conversion command format for the standard configuration</a> .                                                                            |
| R/W                         | Read/Write bit<br>A negated R/W indicates a write configuration command.<br>0 Write                                                                                            |
| ADC_REGISTER_HIGH_BYTE[0:7] | ADC Register High Byte Field<br>REGISTER_HIGH_BYTE is the value to be written into the most significant 8 bits of control/configuration register when the R/W bit is negated.  |
| ADC_REGISTER_LOW_BYTE[0:7]  | ADC Register Low Byte Field<br>REGISTER_LOW_BYTE is the value to be written into the least significant 8 bits of a control/configuration register when the R/W bit is negated. |
| ADC_REG_ADDRESS[0:7]        | ADC Register Address<br>The ADC_REG_ADDRESS field selects a register on the ADC register set to be written or read. Only half-word addresses can be used.                      |

#### 29.6.2.2.1.4 Read configuration command format for on-chip ADC operation

Figure 29-120 describes the command message format for a read configuration command when interfacing with the on-chip ADCs. A read configuration command is used to read the contents of the on-chip ADC registers which are only accessible via command messages. Read configuration commands are differentiated from write configuration commands by an asserted R/W bit.

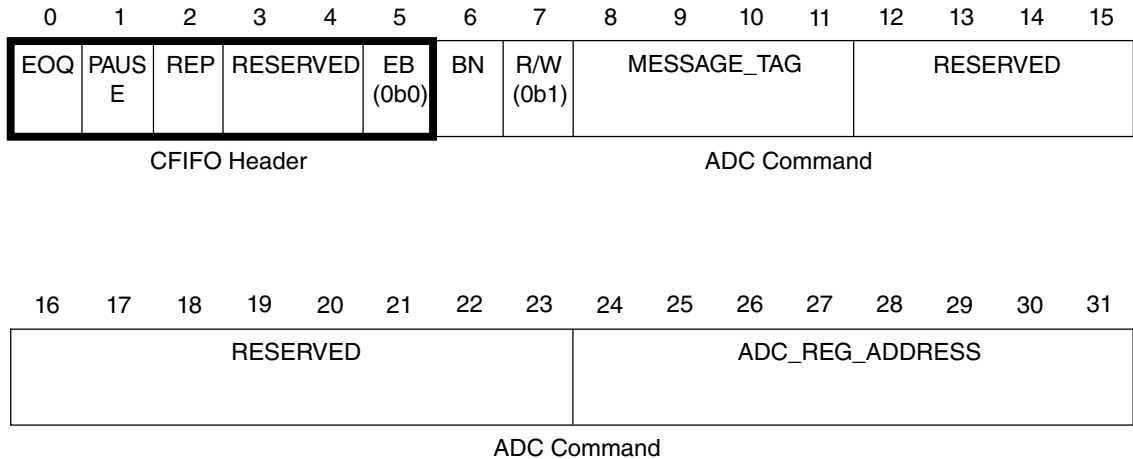


**Figure 29-120. Read configuration command format for on-chip ADC operation**

**Table 29-131. Read configuration command format for on-chip ADC operation — field descriptions**

| Field                | Description                                                                                                                                               |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| EOQ                  | End Of Queue Bit                                                                                                                                          |
| PAUSE                | Pause Bit                                                                                                                                                 |
| REP                  | Repeat/loop Start Point Indication Bit                                                                                                                    |
| BN                   | Buffer Number Bit<br>See <a href="#">Conversion command format for the standard configuration</a> .                                                       |
| R/W                  | Read/Write bit<br>An asserted R/W bit indicates a read configuration command.<br>1 Read                                                                   |
| MESSAGE_TAG[0:3]     | MESSAGE_TAG Field<br>See <a href="#">Conversion command format for the standard configuration</a> ..                                                      |
| ADC_REG_ADDRESS[0:7] | ADC Register Address<br>The ADC_REG_ADDRESS field selects a register on the ADC register set to be written or read. Only half-word addresses can be used. |

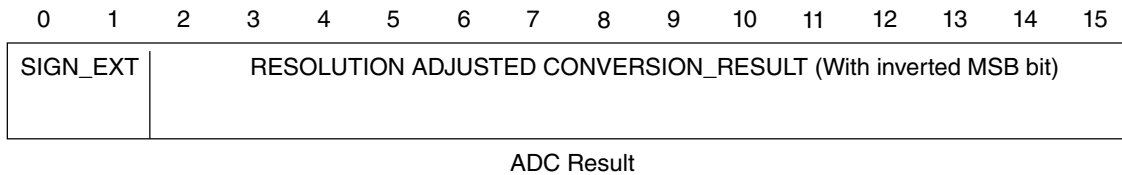
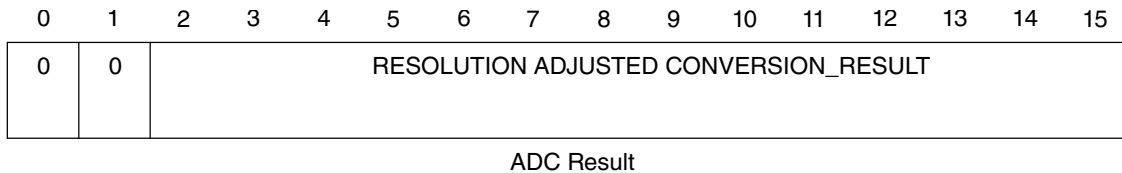
### 29.6.2.2.1.5 ADC result format for on-chip ADC operation

When the *FIFO Control Unit* receives a return data message, it decodes the MESSAGE\_TAG field and stores the 16-bit data into the appropriate RFIFO. This section describes the *ADC result* portion of the *result message* returned by the on-chip ADCs. The 16-bit data stored in the RFIFOs can be:

- Data read from an ADC register with a read configuration command. In this case, the stored 16-bit data corresponds to the contents of the ADC register that was read.
- A time stamp. In this case, the stored 16-bit data is the value of the time base counter latched when the EQADC detects the end of the analog input voltage sampling. For details see [Time stamp feature](#).
- A conversion result, coming directly from the ADCs. In this case, the stored 16-bit data contains a right justified 14-bit result data. The conversion result can be calibrated or not depending on the status of CAL bit in the command that requested the conversion<sup>4</sup>. When the CAL bit is negated, this 14-bit data is obtained by executing a 2-bit left-shift on the 12-bit data resultant from the resolution adjustment on the 8 or 10 or 12-bit data received from the ADC. The resolution adjustment consists of changing the conversion result input from 8, 10 or 12 bits right aligned to a 12-bit word left aligned - see [ADC resolution selection feature](#) for details. When the CAL bit is asserted, this 14-bit data is the result of the calculations performed in the EQADC MAC unit using the 12-bit data result of the resolution adjustment and the calibration constants GCC and OCC, or ALTGCC and ALTOCC - see [ADC calibration feature](#) for details. Then, this 14-bit data is further formatted into a 16-bit format according to the status of the FMT bit in conversion command of the standard configuration or FFMT bit in the conversion command of the alternate configurations<sup>5</sup>. When FMT/FFMT is asserted, the 14-bit result data is reformatted to look as if it was measured against an imaginary ground at VREF/2 (the MSB bit of the 14-bit result is inverted), and is sign-extended to a 16-bit format as in [Figure 29-121](#). When FMT/FFMT is negated, the EQADC zero-extends the 14-bit result data to a 16-bit format as in [Figure 29-122](#). Correspondence between the analog voltage in a channel and the calculated digital values is shown in [Table 29-133](#).

4. In case the conversion result is routed through an on-chip DSP via side interface, the calibration is applied before the data is sent to the DSP.

5. For simplicity, the following text will refer to FMT only, but when using alternate configurations, see [Conversion command format for alternate configurations](#).

**Figure 29-121. ADC Result Format when FMT=1 (Right Justified Signed)****Figure 29-122. ADC result format when FMT=0 (right justified unsigned)****Table 29-132. ADC result formats — field descriptions**

| Field                   | Description                                                                                                                                                                                                                                   |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SIGN_EXT[0:1]           | Sign Extension field<br><br>SIGN_EXT only has meaning when FMT is asserted. SIGN_EXT is 0b00 when CONVERSION_RESULT is positive, and 0b11 when CONVERSION_RESULT is negative.                                                                 |
| CONVERSION_RESULT[0:13] | Conversion Result field<br><br>CONVERSION_RESULT is a digital value corresponding to the analog input voltage in a channel when the conversion command was initiated. The two's complement representation is used to express negative values. |

In [Table 29-133](#) :

- $V_{REF} = V_{RH} - V_{RL} = 5.12V$ . Resulting in one 12-bit count (LSB) = 1.25mV.
- The two's complement representation is used to express negative values.

**Table 29-133. Correspondence between analog voltages and digital values**

| Conversions              | Voltage Level on Channel (V) | Corresponding 8-bit Conversion Result Returned by the ADC | Corresponding 10-bit Conversion Result Returned by the ADC | Corresponding 12-bit Conversion Result Returned by the ADC | 16-bit Result Sent to RFIFOs (FMT=0) <sup>1</sup> | 16-bit Result Sent to RFIFOs (FMT=1) <sup>1</sup> |
|--------------------------|------------------------------|-----------------------------------------------------------|------------------------------------------------------------|------------------------------------------------------------|---------------------------------------------------|---------------------------------------------------|
| Single-Ended Conversions | 5.12                         | -                                                         | -                                                          | 0xFFF                                                      | 0x3FFC                                            | 0x1FFC                                            |
|                          |                              | -                                                         | 0x3FF                                                      | -                                                          | 0x3FF0                                            | 0x1FF0                                            |
|                          |                              | 0xFF                                                      | -                                                          | -                                                          | 0x3FC0                                            | 0x1FC0                                            |
|                          | 5.12 - LSB                   | -                                                         | -                                                          | 0xFFF                                                      | 0x3FFC                                            | 0x1FFC                                            |
|                          |                              | -                                                         | 0x3FF                                                      | -                                                          | 0x3FF0                                            | 0x1FF0                                            |
|                          |                              | 0xFF                                                      | -                                                          | -                                                          | 0x3FC0                                            | 0x1FC0                                            |
|                          | ...                          | ...                                                       | ...                                                        | ...                                                        | ...                                               | ...                                               |

*Table continues on the next page...*

**Table 29-133. Correspondence between analog voltages and digital values (continued)**

| Conversions | Voltage Level on Channel (V) | Corresponding 8-bit Conversion Result Returned by the ADC | Corresponding 10-bit Conversion Result Returned by the ADC | Corresponding 12-bit Conversion Result Returned by the ADC | 16-bit Result Sent to RFIFOs (FMT=0) <sup>1</sup> | 16-bit Result Sent to RFIFOs (FMT=1) <sup>1</sup> |
|-------------|------------------------------|-----------------------------------------------------------|------------------------------------------------------------|------------------------------------------------------------|---------------------------------------------------|---------------------------------------------------|
|             | 2.56                         | -                                                         | -                                                          | 0x800                                                      | 0x2000                                            | 0x0000                                            |
|             |                              | -                                                         | 0x200                                                      | -                                                          | 0x2000                                            | 0x0000                                            |
|             |                              | 0x80                                                      | -                                                          | -                                                          | 0x2000                                            | 0x0000                                            |
|             | ...                          | ...                                                       | ...                                                        | ...                                                        | ...                                               | ...                                               |
|             | 1 LSB                        | -                                                         | -                                                          | 0x001                                                      | 0x0004                                            | 0xE004                                            |
|             |                              | -                                                         | 0x001                                                      | -                                                          | 0x0010                                            | 0xE010                                            |
|             |                              | 0x01                                                      | -                                                          | -                                                          | 0x0040                                            | 0xE040                                            |
|             | 0                            | 0x00                                                      | 0x000                                                      | 0x000                                                      | 0x0000                                            | 0xE000                                            |
|             | Differential Conversions     | 2.56                                                      | -                                                          | -                                                          | 0xFFF                                             | 0x1FFC                                            |
|             |                              |                                                           | -                                                          | 0x3FF                                                      | -                                                 | 0x3FF0                                            |
|             |                              |                                                           | 0xFF                                                       | -                                                          | -                                                 | 0x3FC0                                            |
|             |                              | 2.56 - LSB                                                | -                                                          | -                                                          | 0xFFF                                             | 0x1FFC                                            |
|             |                              |                                                           | -                                                          | 0x3FF                                                      | -                                                 | 0x3FF0                                            |
|             |                              |                                                           | 0xFF                                                       | -                                                          | -                                                 | 0x3FC0                                            |
|             |                              | ...                                                       | ...                                                        | ...                                                        | ...                                               | ...                                               |
|             |                              | 0                                                         | -                                                          | -                                                          | 0x800                                             | 0x2000                                            |
|             |                              |                                                           | -                                                          | 0x200                                                      | -                                                 | 0x2000                                            |
|             |                              |                                                           | 0x80                                                       | -                                                          | -                                                 | 0x2000                                            |
|             |                              | ...                                                       | ...                                                        | ...                                                        | ...                                               | ...                                               |
|             |                              | 2.56 - LSB                                                | -                                                          | -                                                          | 0x001                                             | 0x0004                                            |
|             |                              |                                                           | -                                                          | 0x001                                                      | -                                                 | 0x0010                                            |
|             |                              |                                                           | 0x01                                                       | -                                                          | -                                                 | 0x0040                                            |
|             |                              | -2.56                                                     | 0x00                                                       | 0x000                                                      | 0x000                                             | 0xE000                                            |

1. Assuming uncalibrated conversion results.

### 29.6.3 Command/result queues

The Command and Result queues (CQueues and RQueues) are actually part of the EQADC system although they are not hardware implemented inside the EQADC. Each CQueue entry is a 32-bit Command Message. The last entry of a CQueue has the EOQ bit asserted to indicate that it is the last entry of the CQueue. RQueue entry is a 16-bit data.

See [Overview and basic terminology](#) for a description of the message formats and their flow in EQADC.



See [CQueue and RQueues usage](#) for examples of how CQueues and RQueues can be used.

## 29.6.4 EQADC command FIFOs

### 29.6.4.1 CFIFO basic functionality

There are six prioritized CFIFOs located in the EQADC. Each CFIFO is four entries deep, except CFIFO0 that can be configured to eight entries deep in extended mode, and each CFIFO entry is 32 bits long. A CFIFO serves as a temporary storage location for the command messages stored on the CQueues in the system memory. When a CFIFO is not full, the EQADC sets the corresponding CFFF bit in [EQADC FIFO and Interrupt Status Register \(EQADC\\_FISR<sub>n</sub>\)](#). If CFFE is asserted in [EQADC Interrupt and DMA Control Register 0 \(EQADC\\_IDCR0\)](#), the EQADC generates requests for more commands from a CQueue. An interrupt request, served by the host CPU, is generated when CFFS is negated, and a DMA request, served by the DMAC, is generated when CFFS is asserted. The host CPU or the DMAC respond to these requests by writing to the [EQADC CFIFO Push Register \(EQADC\\_CFPR<sub>n</sub>\)](#) to fill the CFIFO.

#### Note

The DMAC should be configured to write a single command (32-bit data) to the CFIFO push registers for every asserted DMA request it acknowledges. See [EQADC/DMAC interface](#) for DMAC configuration guidelines.

#### Note

CFIFO0 can be configured to work in an alternative way called Streaming Mode. This mode is very different from the mode described here because it maintains some stored commands to execute them several times in sequence and in loop.

#### Note

Only whole words must be written to EQADC\_CFPR. Writing half-words or bytes to EQADC\_CFPR will still push the whole 32-bit CF\_PUSH field into the corresponding CFIFO, but undefined data will fill the areas of CF\_PUSH that were not specifically designated as target locations for writing.

Figure 29-123 describes the important components in the CFIFO. Each CFIFO is implemented as a circular set of registers to avoid the need to move all entries at each push/pop operation. The Push Next Data Pointer points to the next available CFIFO location for storing data written into the EQADC Command FIFO Push Register. The Transfer Next Data Pointer points to the next entry to be removed from CFIFO<sub>x</sub> when it completes a transfer. The *CFIFO Transfer Counter Control Logic* counts the number of entries in the CFIFO and generates DMA or interrupt requests to fill the CFIFO. TNXTPTR in [EQADC FIFO and Interrupt Status Register \(EQADC\\_FISR<sub>n</sub>\)](#) indicates the index of the entry that is currently being addressed by the Transfer Next Data Pointer, and CFCTR, in the same register, provides the number of entries stored in the CFIFO. Using TNXTPTR and CFCTR, the absolute addresses for the entries indicated by the Transfer Next Data Pointer and by the Push Next Data Pointer can be calculated using the following formulas:

Transfer Next Data Pointer Address = CFIFO<sub>x</sub>\_BASE\_ADDRESS + TNXTPTR<sub>x</sub>\*4

Push Next Data Pointer Address = CFIFO<sub>x</sub>\_BASE\_ADDRESS + [(TNXTPTR<sub>x</sub> + CFCTR<sub>x</sub>) mod CFIFO\_DEPTH] \* 4

where

- $a \bmod b$  returns the remainder of the division of  $a$  by  $b$ .
- CFIFO<sub>x</sub>\_BASE\_ADDRESS is the smallest memory mapped address allocated to a CFIFO<sub>x</sub> entry.
- CFIFO\_DEPTH is the number of entries contained in a CFIFO - four in this implementation.

When CFS<sub>x</sub> in [EQADC CFIFO Status Register \(EQADC\\_CFSR\)](#) is TRIGGERED, the EQADC generates the proper control signals for the transfer of the entry pointed by Transfer Next Data Pointer. CFUF<sub>x</sub> in [EQADC FIFO and Interrupt Status Register \(EQADC\\_FISR<sub>n</sub>\)](#) is set when a CFIFO<sub>x</sub> underflow event occurs. A CFIFO underflow occurs when the CFIFO is in TRIGGERED state and it becomes empty. No commands will be transferred from an underflowing CFIFO, nor will command transfers from lower priority CFIFOs be blocked. CFIFO<sub>x</sub> is empty when the Transfer Next Data Pointer  $x$  equals the Push Next Data Pointer  $x$  and CFCTR<sub>x</sub> is zero. CFIFO<sub>x</sub> is full when the Transfer Next Data Pointer  $x$  equals the Push Next Data Pointer  $x$  and CFCTR<sub>x</sub> is not zero.

When the EQADC completes the transfer of an entry from CFIFO<sub>x</sub>: the transferred entry is popped from CFIFO<sub>x</sub>, the CFIFO counter CFCTR in the [EQADC FIFO and Interrupt Status Register \(EQADC\\_FISR<sub>n</sub>\)](#) is decremented by one, and Transfer Next Data Pointer

x is incremented by one (or wrapped around) to point to the next entry in the CFIFO. The transfer of entries bound for the on-chip ADCs is considered completed when they are stored in the appropriate CBuffer.

When the EQADC\_CFPRx is written and CFIFOx is not full, the CFIFO counter CFCTRx is incremented by one, and the Push Next Data Pointer x then is incremented by one (or wrapped around) to point to the next entry in the CFIFO.

When the EQADC\_CFPRx is written but CFIFOx is full, the EQADC will not increment the counter value and will not overwrite any entry in CFIFOx.

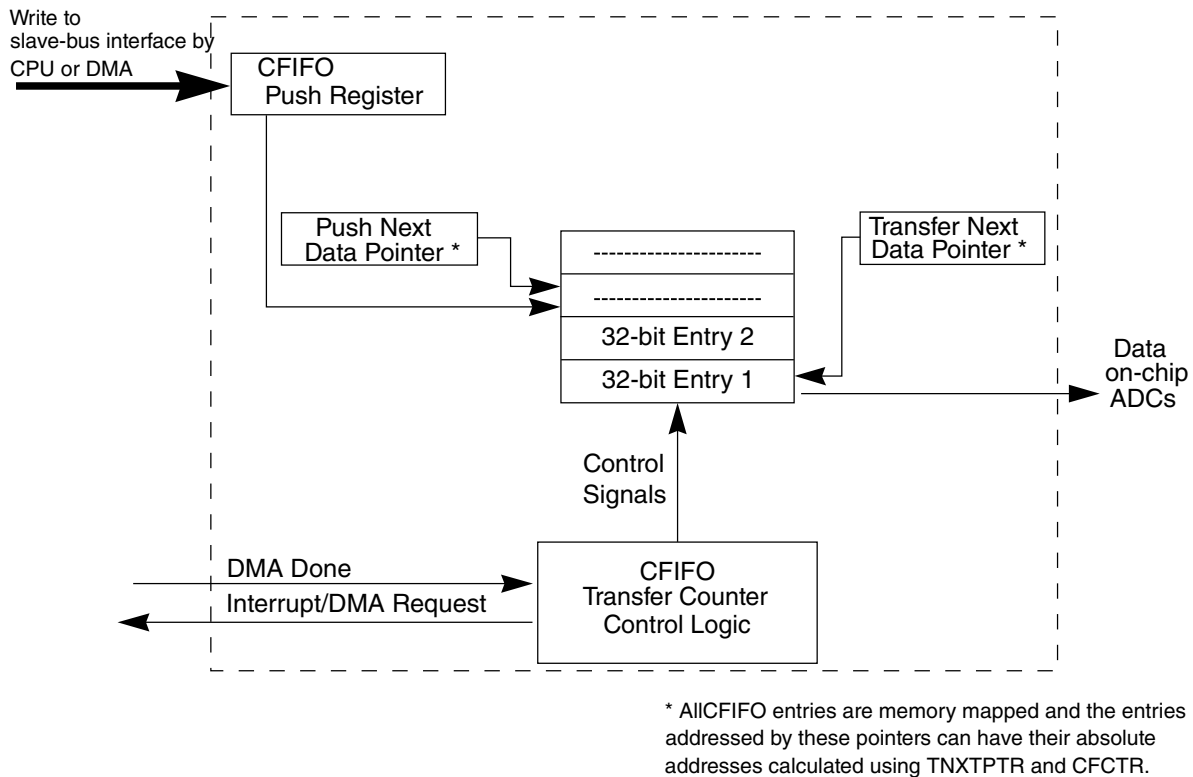


**Figure 29-123. CFIFO Diagram**

The detailed behavior of the Push Next Data Pointer and Transfer Next Data Pointer is described in the example shown in [Figure 29-124](#) where a CFIFO with 16 entries is shown for clarity of explanation, the actual hardware implementation has only four entries. In this example, CFIFOx with 16 entries is shown in sequence after pushing and transferring entries.

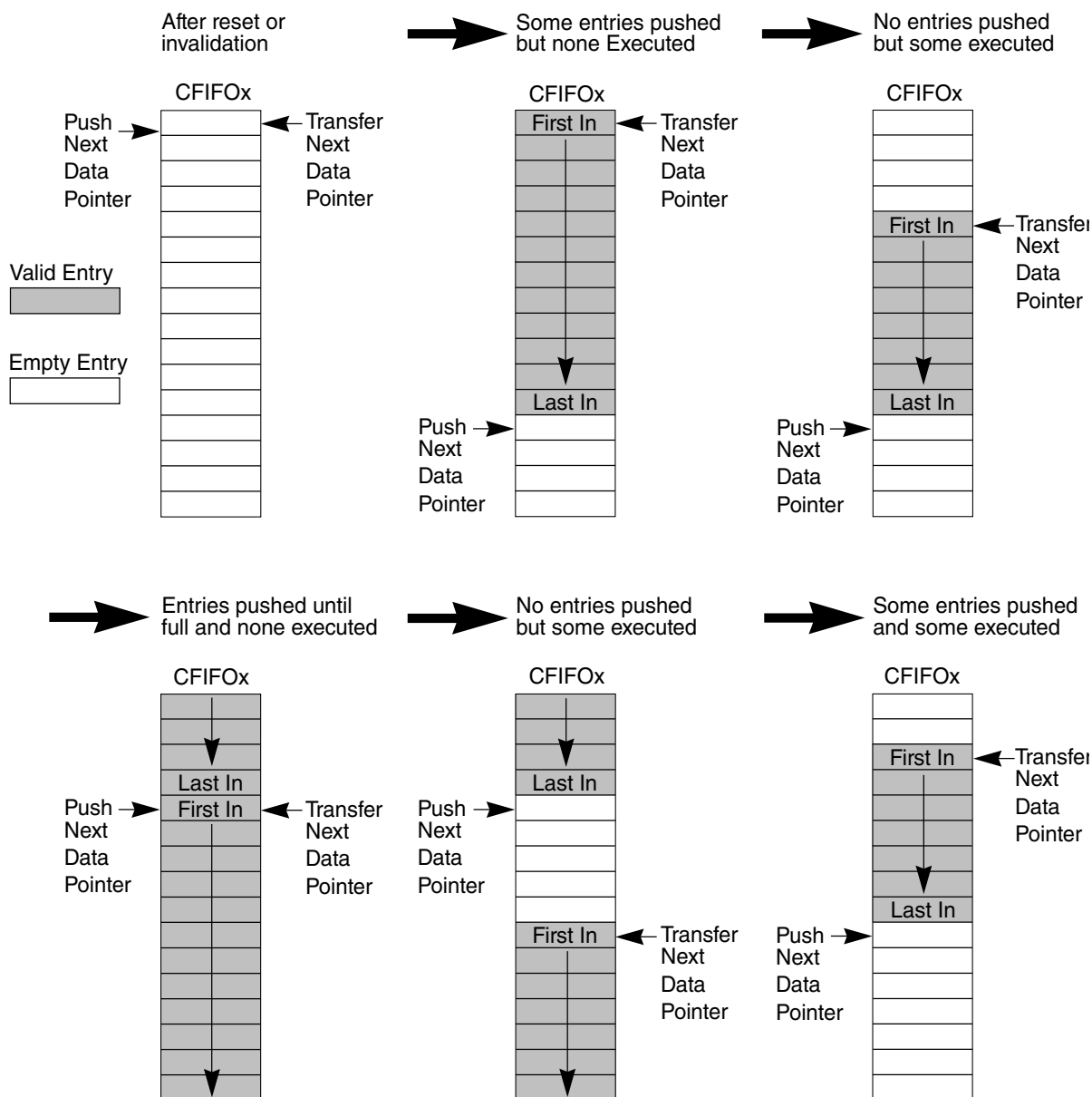


Figure 29-124. CFIFO Entry Pointer Example

### 29.6.4.2 CFIFO0 streaming mode description

CFIFO0 can be configured to operate in streaming mode to allow repetition of a group of commands several times without the need of refilling the registers as in the normal mode of operation of CFIFOs. This mode makes use of the additional bit in the Conversion Command Word (CCW) called 'Repeat' (REP bit). The purpose of this bit is to mark in the command queue, where to start a repeating sequence. This location is stored in an additional pointer 'Repeat Pointer'.

Streaming mode requires 2 trigger inputs. The standard queue 0 trigger, in this mode referred to as Repeat Trigger and a new internal trigger input to the eQADC called Advance Trigger (no filter available).

CFIFO0 is configured to operate in streaming mode by setting the bit STRME0 as described in [EQADC CFIFO Control Register 0 \(EQADC\\_CFCR0\)](#). CFIFO0 is eight entries deep in extended mode by setting the bit CFEEE0 in the same EQADC\_CFCR register, and each entry is 32 bits long. This CFIFO0 serves as a local storage of a few commands that need to be executed sequentially as in a FIFO but can contain sub-queues that need to be executed several times. The CFFF0 bit in [EQADC FIFO and Interrupt Status Register \(EQADC\\_FISR<sub>n</sub>\)](#) is used to assure the CFIFO0 is not full and command messages are stored from address 0x0 to 0x7.

#### 29.6.4.2.1 CFIFO0 operation in streaming mode

In Streaming mode, the CFIFO0 is filled with CCWs using the DMA exactly the same as existing modes. The CFIFO executes commands as per the existing modes until it executes a Conversion Command Word with the Repeat bit set. When this CCW is executed, the Repeat Pointer is set to point to this FIFO location and from this CCW onwards, CFIFO0 entries is not invalidated, that is, the Repeat Pointer prevents this and subsequent entries from being overwritten.

The queue continues to execute until a CCW with an asserted Pause bit is completed; then the queue stops and enters the Pause state, waiting for a trigger. This is the same as normal behavior.

The Pause state is exited in one of two ways: Repeat Trigger or Repeat Trigger with Advance Trigger. The Repeat trigger with no Advance trigger causes the Transfer Next Data Pointer to be loaded with the Repeat Pointer location and CCWs are then executed from the Repeat Pointer back to the Pause bit. This means that a section of the CFIFO0 is repeatedly executed every time a Repeat Trigger occurs.

The Repeat trigger with the Advance trigger pending causes all CCWs from the Repeat pointer to the Pause bit to be invalidated and the CCW after the pause bit to be executed. This is achieved by invalidating the Repeat Pointer. The effect is that the queue advances beyond the repeating section of the CFIFO0 to execute new CCWs.

Note that the Advance trigger can occur at any time between Repeat triggers, but is only granted when the next Repeat trigger occurs. Prior to that it is pending.

In a typical application, the queue is made of some configuration commands to the ADC (to flush the decimator or turn on pad pull-up/down) followed by a repeating section of ADC conversions on one or more ADC channels from one or more sensors; followed by a few more configuration commands; then more repeating ADC conversions, until the

entire engine cycle is complete; when the queue is restarted. The mechanism described permits any number of repeating sub-queues to be loaded and executed, interspersed by configuration commands.

#### 29.6.4.2.2 Triggering description in streaming mode

The additional trigger signal ATRIG0 is detected by a separate circuit that is configured by the bit field AMODE0 as described in [EQADC CFIFO Control Register 0 \(EQADC\\_CFCR0\)](#). This trigger signal is used as an advance control of pop pointer of CFIFO0. In addition, it is used as the enable trigger for the Repeat trigger. This means it is necessary to have an Advance trigger first to enable the detection of the Repeat trigger. When the Repeat trigger is enabled, the Advance trigger is used to advance the pop pointer beyond some loop sub-queue. And it is used to disable the Repeat trigger by executing a Pause without a previous REP bit. [Figure 29-125](#) and [Figure 29-126](#) illustrate these two triggers dependency.

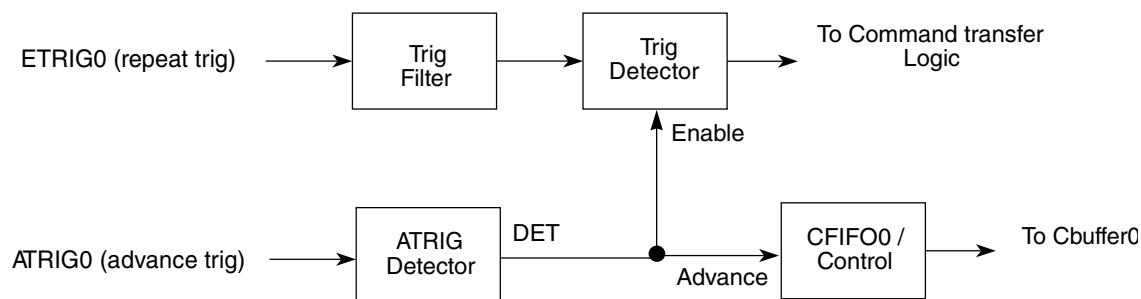


**Figure 29-125. Advance trigger and repeat trigger relationship Block Diagram**

In Streaming mode, the CFIFO0 is filled with CCWs using the DMA as usual. The two triggers are configured to positive edge and single scan mode.

The SSS bit is asserted and the trigger detector of the Repeat trigger is disabled in the start of the queue. It is necessary to receive the first Advance trigger to enable the detector of the other trigger. This enable is useful when the Repeat trigger is received all the time and the trigger signal can be disabled when it is not desired.

The Advance trigger is received and detected and the Repeat trigger detector is enabled. No commands are executed until now.

The Repeat trigger is detected and the commands start to be executed in sequence. If a REP bit is decoded with the PAUSE bit, the loop is configured and the CFIFO0 commands stop to be executed. The next Repeat trigger is waited to start the execution of the loop again, or the Advance trigger can be detected to break the loop and advance the queue in CFIFO0. The Repeat trigger detector remains enabled.

If the Advance trigger is received and the next command in the CFIFO0 does not present the REP bit set, this means the CFIFO0 is not starting a new loop. In this case (outside a loop) if a PAUSE bit is decoded, this means to disable the Repeat trigger detector. This can be useful if the Repeat trigger is not required for some interval of time. The Repeat trigger detector is enabled again when the next Advance trigger event is detected.

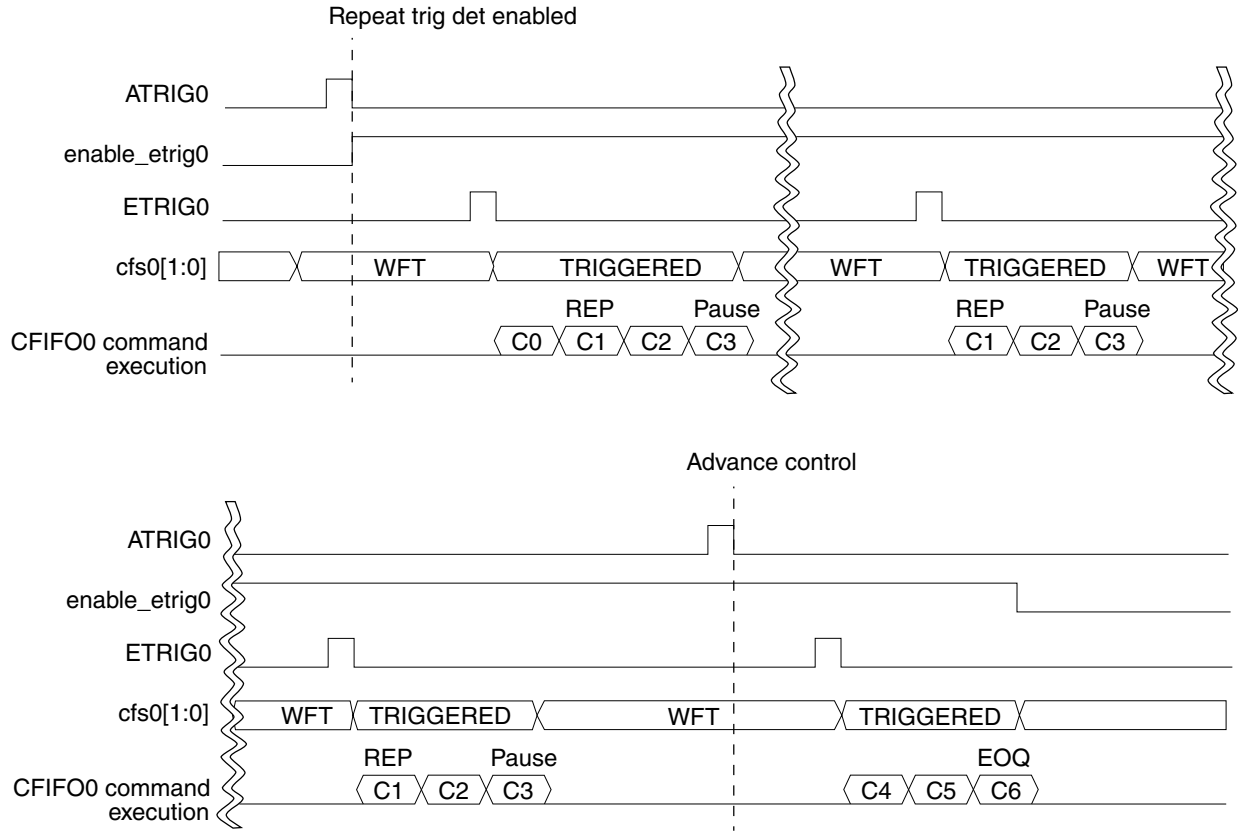


**Figure 29-126. Timing diagram of typical use of Advance trigger and Repeat trigger**

#### 29.6.4.2.3 CFIFO0 diagram description in streaming mode

Figure 29-127 represents the main components of CFIFO0 in streaming mode. However, some signals behave in a different way from the common operation. The Push Next Data Pointer points to the next available CFIFO0 location for storing data written into the EQADC Command FIFO Push Register. The Transfer Next Data Pointer points to the next entry to be transferred to Cbuffer. The Repeat Pointer points to the first entry of the repeating sub-queue. TNXTPTR in [EQADC FIFO and Interrupt Status Register \(EQADC\\_FISR<sub>n</sub>\)](#) indicates the index of the entry that is currently being addressed by the Transfer Next Data Pointer, and CFCTR, in the same register, provides the number of entries stored in the CFIFO.



When CFS0 in [EQADC CFIFO Status Register \(EQADC\\_CFSR\)](#) is TRIGGERED, the EQADC generates the proper control signals for the transfer of the entry pointed by Transfer Next Data Pointer. CFUF0 in [EQADC FIFO and Interrupt Status Register \(EQADC\\_FISR<sub>n</sub>\)](#) is set when CFIFO0 underflow event occurs. A CFIFO underflow occurs when the CFIFO is in TRIGGERED state and it is empty. No commands will be transferred from an underflowing CFIFO, nor will command transfers from lower priority CFIFOs be blocked. CFIFO0 is empty when CFCTR0 is zero. CFIFO0 is full when (CFCTR0 mod CFIFO\_DEPTH) is zero but CFCTR0 is not zero.

When the EQADC completes the transfer of an entry from CFIFO0 in loop condition: the transferred entry is not popped from CFIFO0, the CFIFO counter CFCTR in the [EQADC FIFO and Interrupt Status Register \(EQADC\\_FISR<sub>n</sub>\)](#) is not decremented by one, and Transfer Next Data Pointer 0 is incremented by one (or wrapped around) to point to the next entry in the CFIFO0.

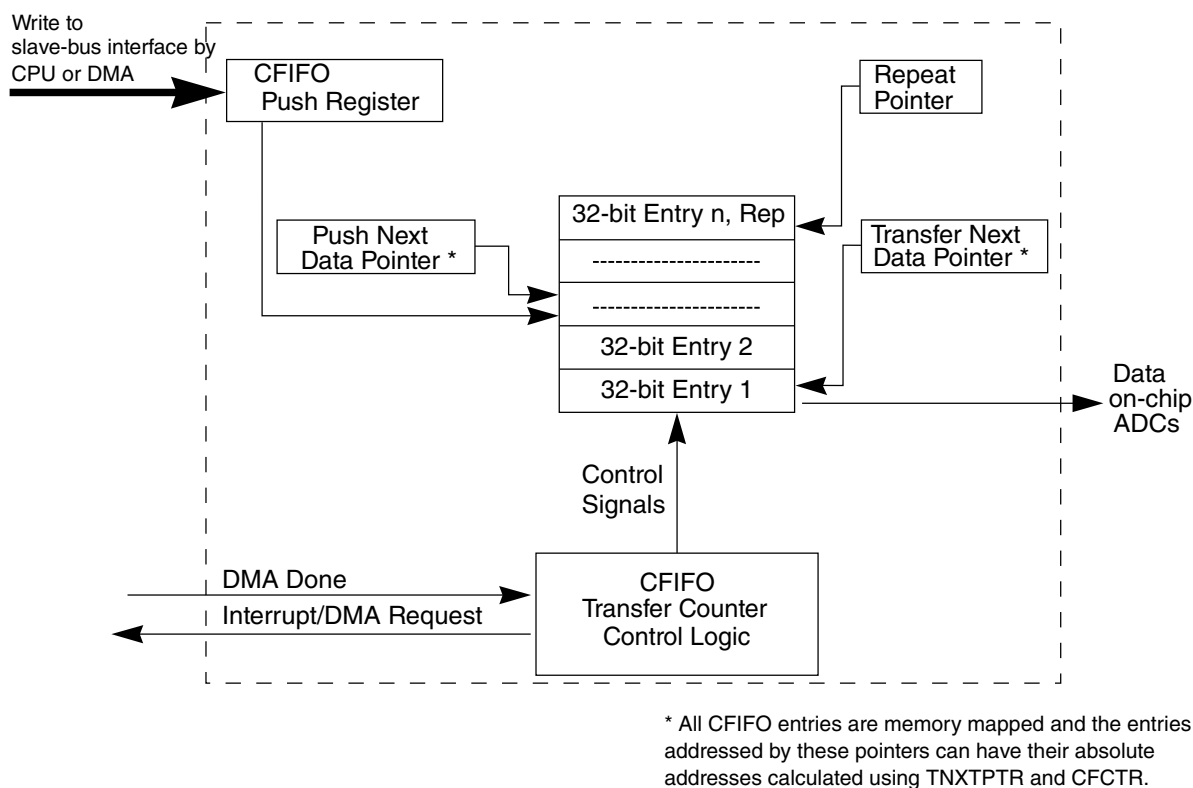


**Figure 29-127. CFIFO0 in streaming mode diagram**

The detailed behavior of the Push Next Data Pointer and Transfer Next Data Pointer is described in the example shown in [Figure 29-128](#) where a CFIFO with 16 entries is shown for clarity of explanation, the actual hardware implementation has only four/eight entries. In this example, CFIFO0 with 16 entries is shown in sequence after pushing and transferring entries.



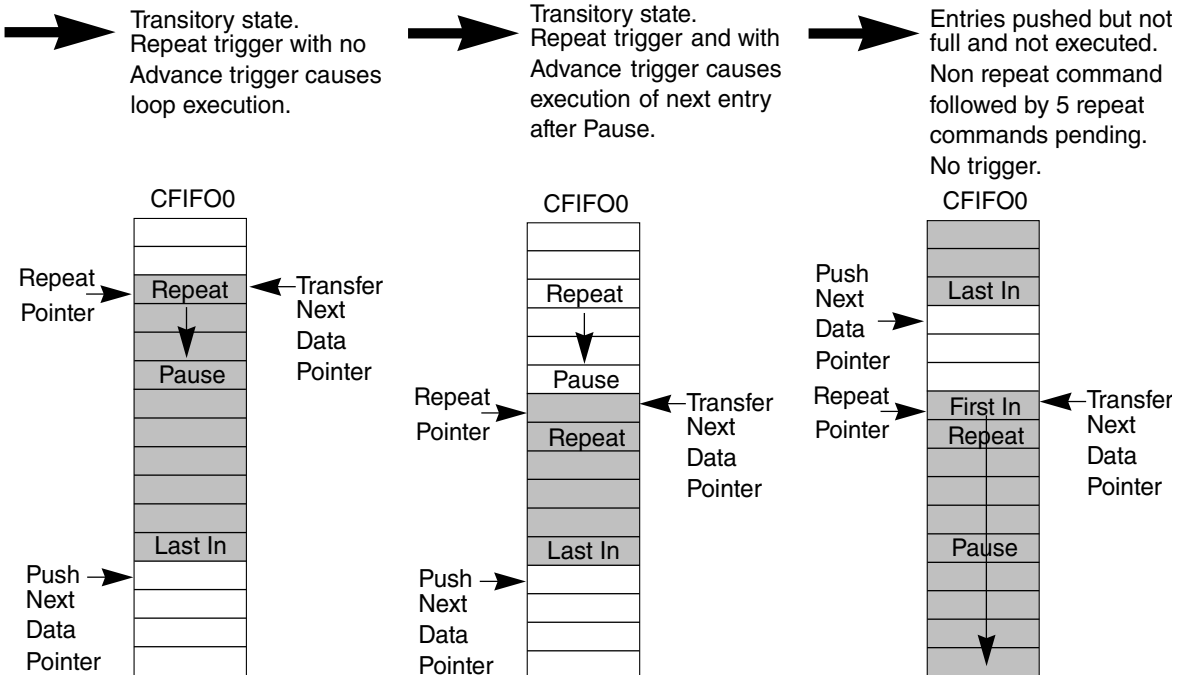


**Figure 29-128. CFIFO0 in streaming mode entry pointer example**

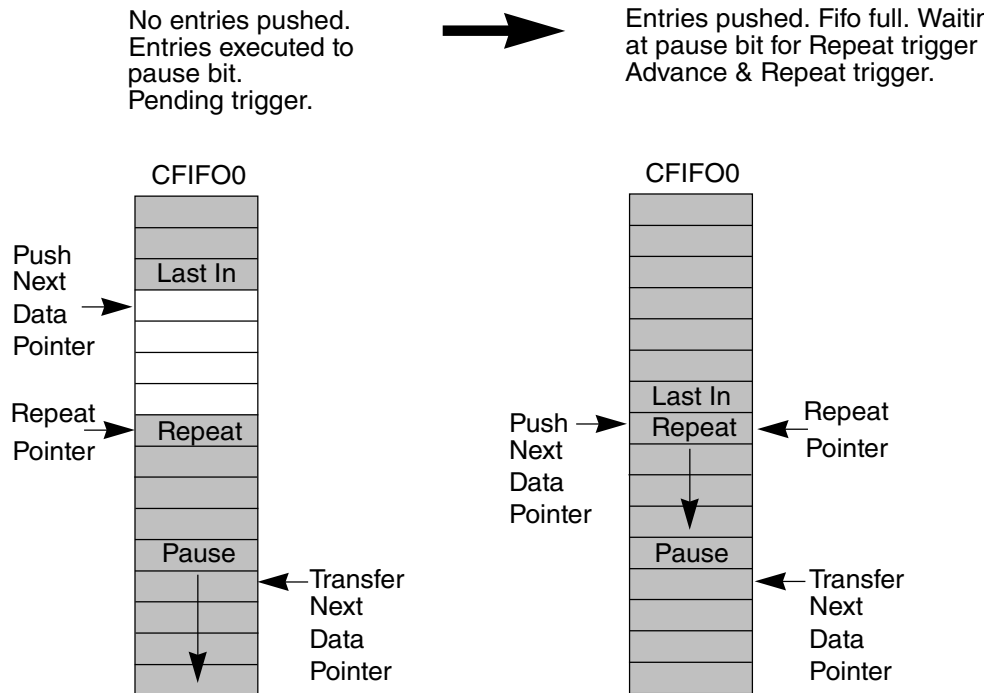


Figure 29-129. CFIFO0 in streaming mode entry pointer example (continued)

#### 29.6.4.2.4 Streaming mode error conditions

In the repeat state, the existing error conditions still apply, but now there are new ways to trigger them. Now, the CCWs are not being invalidated so the DMA is not able to load more CCWs into those locations. So a queue overflow becomes more likely, and occurs if the repeat loop is longer than 8 entries. If all CCWs in the CFIFO0 are executed and no Pause bit or EOQ bit is detected, the eQADC will signal an underflow error. In practice this may limit a repeating queue to 7 entries since otherwise an underflow will occur at the point a Repeat with Advance trigger occurs, and there is no command in the CFIFO0 to execute. The exception is a final command with both a Pause and an EOQ bit set. The End of Queue bit EOQ continues to operate as in normal mode, unless the Repeat mode is enabled. In this case the Pause bit takes precedence and a Repeat trigger causes the jump back described. A Repeat trigger with Advance trigger causes the queue to end.

Another error condition occur when the repeat trigger is in the TRIGGERED state and a new repeat trigger is received. In this case, a trigger overflow occurs but the CFIFO0 is defined to not restart the loop. The trigger in this case is not used in the CFIFO0, but the overflow is indicated.

### 29.6.4.3 CFIFO common prioritization and command transfer

The CFIFO priority is fixed according to the CFIFO number. A CFIFO with a smaller number has a higher priority. When commands of distinct CFIFOs are bound for the same destination (CBuffer), the higher priority CFIFO is always served first. A TRIGGERED, not-underflowing CFIFO will start the transfer of its commands when:

- its commands are bound for an internal CBuffer that is not full, and it is the highest priority triggered CFIFO sending commands to that CBuffer.

A triggered CFIFO with commands bound for a certain CBuffer consecutively transfers its commands to it until:

- an asserted End Of Queue bit is reached, or;
- an asserted Pause bit is encountered and the CFIFO is configured for edge trigger mode, or;
- CFIFO is configured for level trigger mode and a closed gate is detected, or;
- in case its commands are bound for an internal CBuffer, a higher priority CFIFO that uses the same internal CBuffer is triggered

The prioritization logic of the EQADC, depicted in [Figure 29-130](#), is composed of three independent sub-blocks: one prioritizing CFIFOs with commands bound for CBuffer0, another prioritizing CFIFOs with commands for CBuffer1. As these two sub-blocks are independent, simultaneous writes to CBuffer0, to CBuffer1 are allowed. The hardware identifies the destination of a command by decoding the BN bit in the command message - see [Message format in EQADC](#) for details.

#### Note

Triggered but empty CFIFOs, underflowing CFIFOs, are not considered for prioritization. No data from these CFIFOs will be sent to the CBuffers and nor will they stop lower priority CFIFOs from transferring commands.

Whenever CBuffer0 is able to receive new entries, the prioritization sub-block selects the highest-priority triggered CFIFO with a command bound for CBuffer0, and writes its command into the buffer. In case CBuffer0 is able to receive new entries but there are no triggered CFIFOs with commands bound for it, nothing is written to the buffer. The sub-block prioritizing CBuffer1 usage behaves in the same way.

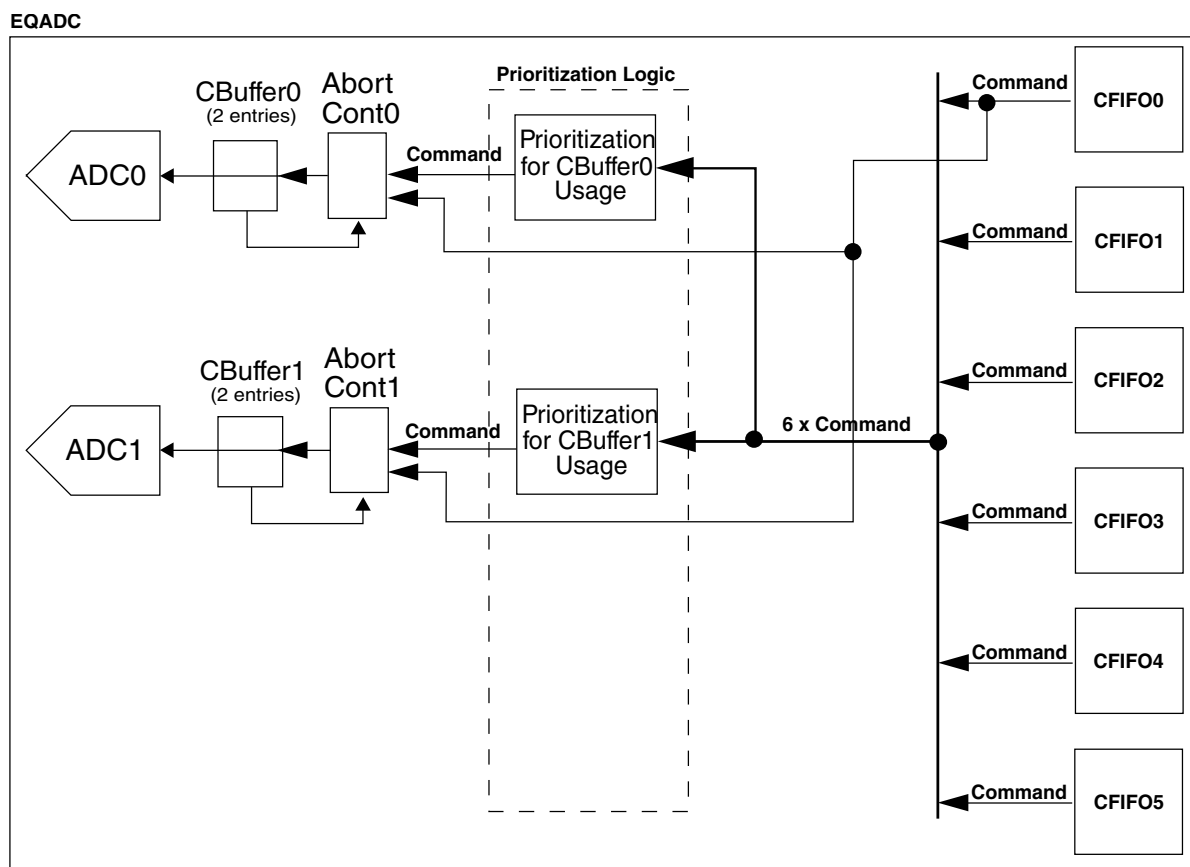


Figure 29-130. CFIFO prioritization logic

#### 29.6.4.4 CFIFO prioritization in abort mode

The CFIFO priority does not change when the EQADC is configured to allow abortion of conversion execution in on-chip ADC analog blocks. However, CFIFO0 is the only one that can be enabled to abort conversions.

This feature is necessary when the timing of some conversion is very important. In normal priority scheme, when CFIFO0 is triggered, its conversion command can be put behind 2 pending conversion commands in the Cbuffer due to the queue structure. Considering that these 2 pending commands are from lower priority CFIFOs and that the delay between the trigger and the sampling of the command from CFIFO0 can be unacceptable, EQADC can be configured to permit immediate conversion commands from CFIFO0 with abort function.

When CFIFO0 is triggered and abort is enabled, up to 2 commands in Cbuffer0 or Cbuffer1 are stored in a side register. The abort request signal is generated to ADC0 or ADC1 and the confirmation of ADC reset/ready is waited to send the command from CFIFO0 to the decoded Cbuffer.

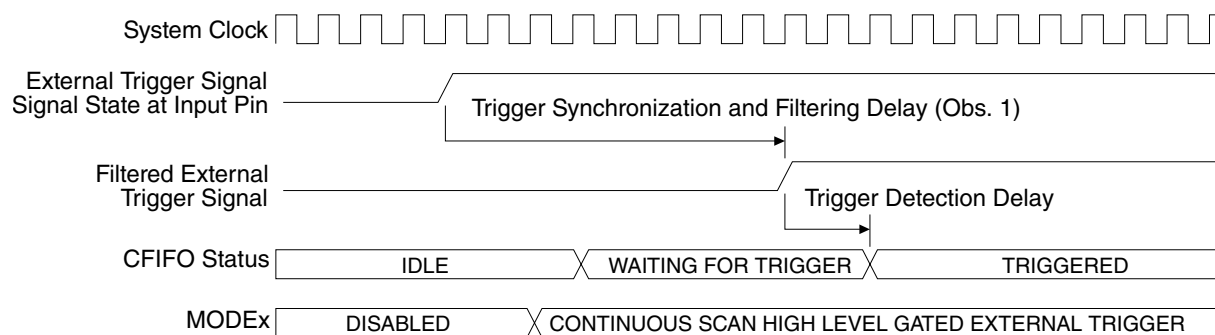
After the transfer of all commands from CFIFO0, the recovery phase restores the up to 2 commands that were in Cbuffer when the abort occurred. After this recovery phase, it is established the normal process of prioritization of commands from CFIFOs.

### 29.6.4.5 External trigger event detection

The digital filters for trigger signals can be individually bypassed by asserting the input control signals eqadc\_intern\_trig\_sel5-0. When the filter is bypassed, the ETRIG input signal is not filtered and the logic after the filter receives a copy of this input trigger signal.

The Digital Filter Length field in [EQADC External Trigger Digital Filter Register \(EQADC\\_ETDFR\)](#) specifies the minimum number of system clocks that the ETRIG0-5 signals must be held at a logic level to be recognized as valid. All ETRIG signals are filtered. A counter for each queue trigger is implemented to detect a transition between logic levels. The counter counts at the system clock rate. The corresponding counter is cleared and restarted each time the signal transitions between logic levels. When the corresponding counter matches the value specified by the Digital Filter Length field in [EQADC External Trigger Digital Filter Register \(EQADC\\_ETDFR\)](#), the EQADC considers the ETRIG logic level to be valid and passes that new logic level to the rest of the EQADC.

The filter is only for filtering the ETRIG signal. Logic after the filter checks for transitions between filtered values, such as for detecting the transition from a filtered logic level zero to a filter logic level one in rising edge external trigger mode. The EQADC can detect rising edge, falling edge, or level gated external triggers. The digital filter will always be active independently of the status of the MODEx field in [EQADC CFIFO Control Register 0 \(EQADC\\_CFCR0\)](#), but the edge, level detection logic is only active when MODEx is set to a value different from disabled, and in case MODEx is set to single scan mode, when the SSS bit is asserted. Note that the time necessary for an external trigger event to result into a CFIFO status change is not solely determined by the DFL field in the [EQADC External Trigger Digital Filter Register \(EQADC\\_ETDFR\)](#). After being synchronized to the system clock and filtered, a trigger event is checked against the CFIFO trigger mode. Only then, after a valid trigger event is detected, the EQADC accordingly changes the CFIFO status. See [Figure 29-131](#) for an example.



Obs.

- 1: This delay is about 2 clocks when the filter bypass control is asserted.

**Figure 29-131. ETRIG event propagation example**

### 29.6.4.6 CFIFO scan trigger modes

The EQADC supports two different scan modes, single-scan and continuous-scan. Refer to [Table 29-134](#) for a summary of these two scan modes. When a CFIFO is triggered, the EQADC scan mode determines whether the EQADC will stop command transfers from a CFIFO, and wait for software intervention to rearm the CFIFO to detect new trigger events, upon detection of an asserted EOQ bit in the last transfer. See [Message format in EQADC](#) for details about command formats.

CFIFOs can be configured in single-scan or continuous-scan mode. When a CFIFO is configured in single-scan mode, the EQADC scans the CQueue one time. The EQADC stops future command transfers from the triggered CFIFO after detecting the EOQ bit set in the last transfer. After a EOQ bit is detected, software involvement is required to rearm the CFIFO so that it can detect new trigger events.

When a CFIFO is configured for continuous-scan mode, no software involvement is necessary to rearm the CFIFO to detect new trigger events after an asserted EOQ is detected. In continuous-scan mode the whole CQueue is scanned multiple times.

The EQADC also supports different triggering mechanisms for each scan mode. The EQADC will not transfer commands from a CFIFO until the CFIFO is triggered. The combination of scan modes and triggering mechanisms allows the support of different requirements for scanning input channels. The scan mode and trigger mechanism are configured by programming the MODEx field in [EQADC CFIFO Control Register 0 \(EQADC\\_CFCR0\)](#).

Enabled CFIFOs can be triggered by software or external trigger events. The elapsed time from detecting a trigger to transferring a command is a function of clock frequency, trigger synchronization, trigger filtering or not, programmable trigger events, command

transfer, CFIFO prioritization, CBuffer availability, etc. Fast and predictable transfers can be achieved by ensuring that the CFIFO is not underflowing and that the target CBuffer is not full when the CFIFO is triggered.

#### 29.6.4.6.1 Disabled Mode

The MODEx field in [EQADC CFIFO Control Register 0 \(EQADC\\_CFCR0\)](#) for all of the CFIFOs can be changed from any other mode to disabled at any time. No trigger event can initiate command transfers from an CFIFO which has its MODE field programmed to disabled.

#### Note

If MODEx is not disabled, it must not be changed to any other mode besides disabled. If MODEx is disabled and the CFIFO status is IDLE, MODEx can be changed to any other mode.

If MODEx is changed to disabled:

- The CFIFO execution status will change to IDLE. The timing of this change depends on whether a command is being transferred or not:
  - When no command transfer is in progress, the EQADC switches the CFIFO to IDLE status immediately.
  - When a command transfer to an on-chip CBuffer is in progress, the EQADC will complete the transfer, update TC\_CF, and switch CFIFO status to IDLE. Command transfers to the internal CBuffers are considered completed when a command is written to the buffers.
- The CFIFOs are not invalidated automatically. The CFIFO still can be invalidated by writing a "1" to the CFINVx bit in [EQADC CFIFO Control Register 0 \(EQADC\\_CFCR0\)](#). Certify that CFS has changed to IDLE before setting CFINVx.
- The TC\_CFx value also is not reset automatically, but it can be reset by writing "0" to it.
- The SSS bit in [EQADC FIFO and Interrupt Status Register \(EQADC\\_FISRn\)](#) is negated. The SSS bit can be set even if a "1" is written to the SSE bit in [EQADC CFIFO Control Register 0 \(EQADC\\_CFCR0\)](#) in the same write that the MODEx field is changed to a value other than disabled.
- The trigger detection hardware is reset. If MODEx is changed from disabled to an edge trigger mode, a new edge, matching that edge trigger mode, is needed to trigger the command transfers from the CFIFO.

## Note

CFIFO fill requests, which generated when CFFF is asserted, are not automatically halted when MODE<sub>x</sub> is changed to disabled. CFIFO fill requests will still be generated until CFFE is cleared in [EQADC Interrupt and DMA Control Register 0 \(EQADC\\_IDCR0\)](#).

### 29.6.4.6.2 Single-scan mode

In single-scan mode, a single pass through a sequence of command messages in a CQueue is performed.

In single-scan software trigger mode, the CFIFO is triggered by an asserted Single-Scan Status bit (SSS) in [EQADC FIFO and Interrupt Status Register \(EQADC\\_FISR<sub>n</sub>\)](#). The SSS bit is set by writing "1" to the Single-Scan Enable bit (SSE) in [EQADC CFIFO Control Register 0 \(EQADC\\_CFCR0\)](#).

In single-scan edge- or level-trigger mode, the respective triggers are only detected when the SSS bit is asserted. When the SSS bit is negated, all trigger events for that CFIFO are ignored. Writing a "1" to the SSE bit can be done during the same write cycle that the CFIFO operation mode is configured.

Only the EQADC can clear the SSS bit. Once SSS is asserted, it remains asserted until the EQADC completes the CQueue scan, or the CFIFO operation mode (MODE<sub>x</sub>) in [EQADC CFIFO Control Register 0 \(EQADC\\_CFCR0\)](#) is changed to disabled. The SSS<sub>x</sub> bit will be negated while MODE<sub>x</sub> is disabled.

#### 29.6.4.6.2.1 Single-Scan Software Trigger

When single-scan software trigger mode is selected, the CFIFO is triggered by an asserted SSS bit. The SSS bit is asserted by writing "1" to the SSE bit. Writing to SSE while SSS is already asserted will not have any effect on the state of the SSS bit, nor will it cause a trigger overrun event.

The CFIFO commands start to be transferred when the CFIFO becomes the highest priority CFIFO using a not-full on-chip CBuffer. When an asserted EOQ bit is encountered, the EQADC will clear the SSS bit. Setting the SSS bit is required for the EQADC to start the next scan of the queue.

The Pause bit has no effect in single-scan software trigger mode.



#### 29.6.4.6.2.2 Single-scan edge trigger

When SSS is asserted and an edge triggered mode is selected for a CFIFO, an appropriate edge on the associated trigger signal causes the CFIFO to become TRIGGERED. For example, if rising-edge trigger mode is selected, the CFIFO becomes TRIGGERED when a rising edge is sensed on the trigger signal. The CFIFO commands start to be transferred when the CFIFO becomes the highest priority CFIFO using a not-full on-chip CBuffer.

When an asserted EOQ bit is encountered, the EQADC clears SSS and stops command transfers from the CFIFO. An asserted SSS bit and a subsequent edge trigger event are required to start the next scan for the CFIFO. When an asserted Pause bit is encountered, the EQADC stops command transfers from the CFIFO, but SSS remains set. Another edge trigger event is required for command transfers to continue. A trigger overrun happens when the CFIFO is in TRIGGERED state and an edge trigger event is detected.

#### 29.6.4.6.2.3 Single-scan level trigger

When SSS is asserted and a level gated trigger mode is selected, the input level on the associated trigger signal puts the CFIFO in TRIGGERED state. When the CFIFO is asserted to high-level gated trigger, a high level signal opens the gate, and a low level closes the gate. When the CFIFO is set to low-level gated trigger mode, a low level signal opens the gate, and a high level closes the gate. If the corresponding level is already present, setting the SSS bit triggers the CFIFO. The CFIFO commands start to be transferred when the CFIFO becomes the highest priority CFIFO using a not-full on-chip CBuffer.

The EQADC clears the SSS bit and stops transferring commands from a TRIGGERED CFIFO when an asserted EOQ bit is encountered or when CFIFO status changes from TRIGGERED due to the detection of a closed gate. If a closed gate is detected while no command transfers are taking place and the CFIFO status is TRIGGERED, the CFIFO status is immediately changed to IDLE, the SSS bit is negated, and the PF flag is asserted. Once the transmission is completed, the TC\_CF counter is updated, the SSS bit is negated, the PF flag is asserted, and the CFIFO status is changed to IDLE. An asserted SSS bit and a level trigger are required to restart the CFIFO. Command transfers will restart from the point they have stopped.

The Pause bit has no effect in single-scan level-trigger mode.

### 29.6.4.6.3 Continuous-scan mode

In continuous-scan mode, multiple passes looping through a sequence of command messages in a CQueue are executed. When a CFIFO is programmed for a continuous-scan mode, the SSE bit in the [EQADC CFIFO Control Register 0 \(EQADC\\_CFCR0\)](#) does not have any effect.

#### 29.6.4.6.3.1 Continuous-scan software trigger

When a CFIFO is programmed to continuous-scan software trigger mode, the CFIFO is triggered immediately. The CFIFO commands start to be transferred when the CFIFO becomes the highest priority CFIFO using a not-full on-chip CBuffer. When a CFIFO is programmed to run in continuous-scan software trigger mode, the EQADC will not halt transfers from the CFIFO until the CFIFO operation mode is modified to disabled or a higher priority CFIFO preempts it. Although command transfers will not stop upon detection of an asserted EOQ bit, the EOQF is set and, if enabled, an EOQ interrupt request is generated.

The Pause bit has no effect in continuous-scan software trigger mode.

#### 29.6.4.6.3.2 Continuous-scan edge trigger

When rising, falling, or either edge trigger mode is selected for a CFIFO, a corresponding edge on the associated ETRIG signal places the CFIFO in TRIGGERED state. The CFIFO commands start to be transferred when the CFIFO becomes the highest priority CFIFO using a not-full on-chip CBuffer.

When an EOQ or a Pause is encountered, the EQADC halts command transfers from the CFIFO and, if enabled, the appropriate interrupt requests are generated. Another edge trigger event is required to resume command transfers but no software involvement is required to rearm the CFIFO in order to detect such event.

A trigger overrun happens when the CFIFO is already in TRIGGERED state and a new edge trigger event is detected.

#### 29.6.4.6.3.3 Continuous-scan level trigger

When high or low level gated trigger mode is selected, the input level on the associated trigger signal places the CFIFO in TRIGGERED state. When high-level gated trigger is selected, a high-level signal opens the gate, and a low level closes the gate. The CFIFO commands start to be transferred when the CFIFO becomes the highest priority CFIFO.

using a not-full on-chip CBuffer. Although command transfers will not stop upon detection of an asserted EOQ bit at the end of a command transfer, the EOQF is asserted and, if enabled, an EOQ interrupt request is generated.

The EQADC stops transferring commands from a TRIGGERED CFIFO when CFIFO status changes from TRIGGERED due to the detection of a closed gate. If a closed gate is detected while no command transfers are taking place and the CFIFO status is TRIGGERED, the CFIFO status is immediately changed to WAITING FOR TRIGGER and the PF flag is asserted. Once the transmission is completed, the TC\_CF counter is updated, the PF flag is asserted, and the CFIFO status is changed to WAITING FOR TRIGGER. Command transfers will restart as the gate opens.

The Pause bit has no effect in continuous-scan level-trigger mode.

#### 29.6.4.6.4 CFIFO scan trigger mode start/stop summary

Table 29-134 summarizes the start and stop conditions of command transfers from CFIFOs for all of the single-scan and continuous-scan trigger modes.

**Table 29-134. CFIFO scan trigger mode - command transfer start/stop summary**

| Trigger mode             | Requires asserted SSS to recognize trigger events? | Command transfer start/restart condition                          | Stop on asserted EOQ bit <sup>1</sup> ? | Stop on asserted Pause bit <sup>2</sup> ? | Other command transfer stop condition <sup>3,4</sup>                                                                                   |
|--------------------------|----------------------------------------------------|-------------------------------------------------------------------|-----------------------------------------|-------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| Single Scan Software     | Don't Care                                         | Asserted SSS bit.                                                 | Yes                                     | No                                        | None.                                                                                                                                  |
| Single Scan Edge         | Yes                                                | A corresponding edge occurs when the SSS bit is asserted.         | Yes                                     | Yes                                       | None.                                                                                                                                  |
| Single Scan Level        | Yes                                                | Gate is opened when the SSS bit is asserted.                      | Yes                                     | No                                        | EQADC also stops transfers from the CFIFO when CFIFO status changes from TRIGGERED due to the detection of a closed gate. <sup>5</sup> |
| Continuous Scan Software | No                                                 | CFIFO starts automatically after being configured into this mode. | No                                      | No                                        | None.                                                                                                                                  |
| Continuous Scan Edge     | No                                                 | A corresponding edge occurs.                                      | Yes                                     | Yes                                       | None.                                                                                                                                  |
| Continuous Scan Level    | No                                                 | Gate is opened.                                                   | No                                      | No                                        | EQADC also stops transfers from the CFIFO when CFIFO status changes from TRIGGERED due to the detection of a closed gate. <sup>5</sup> |

1. See [CQueue completion status](#) for more information on EOQ.

## CFIFO and trigger status

2. See [Pause status](#) for more information on Pause.
3. EQADC always stops command transfers from a CFIFO when the CFIFO operation mode is disabled.
4. EQADC always stops command transfers from a CFIFO when a higher priority CFIFO is triggered. See [CFIFO common prioritization and command transfer](#) for information on CFIFO priority.
4. If a closed gate is detected while no command transfers are taking place, it will have immediate effect on the CFIFO status.

### 29.6.4.7 CFIFO and trigger status

#### 29.6.4.7.1 CFIFO operation status

Each CFIFOs has its own CFIFO status field. CFIFO status (CFS) can be read from [EQADC CFIFO Status Register \(EQADC\\_CFSR\)](#). [Figure 29-132](#) and [Table 29-135](#) indicate the CFIFO status switching condition. See [Table 29-42](#) for the meaning of each CFIFO operation status. The last CFIFO to transfer a command to an on-chip CBuffer can be read from the LCFTCB $n$  ( $n=0,1$ ) fields in [EQADC CFIFO Status Snapshot Register 0 \(EQADC\\_CFSSR0\)](#).

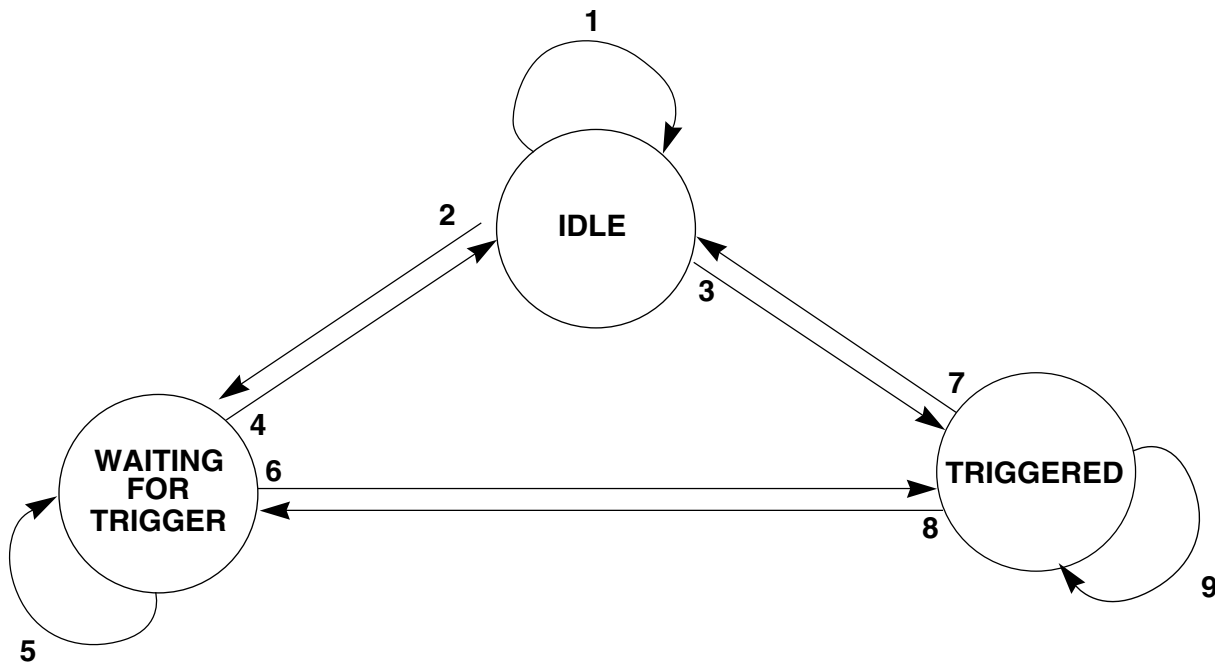


Figure 29-132. State machine of CFIFO status

Table 29-135. Command FIFO status switching condition

| No. | From current CFIFO status (CFS) | To new CFIFO status (CFS) | Status switching condition                                                                                                                                                                  |
|-----|---------------------------------|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1   | IDLE (00)                       | IDLE (0b00)               | <ul style="list-style-type: none"><li>• CFIFO Mode is programmed to disabled, OR</li><li>• CFIFO Mode is programmed to single-scan edge or level trigger mode and SSS is negated.</li></ul> |

Table continues on the next page...

**Table 29-135. Command FIFO status switching condition  
(continued)**

| No. | From current CFIFO status (CFS) | To new CFIFO status (CFS)  | Status switching condition                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-----|---------------------------------|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2   |                                 | WAITING FOR TRIGGER (0b10) | <ul style="list-style-type: none"> <li>CFIFO Mode is programmed to continuous-scan edge or level trigger mode, OR</li> <li>CFIFO Mode is programmed to single-scan edge or level trigger mode and SSS is asserted, OR</li> <li>CFIFO Mode is programmed to single-scan software trigger mode.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 3   |                                 | TRIGGERED (0b11)           | <ul style="list-style-type: none"> <li>CFIFO Mode is programmed to continuous-scan software trigger mode</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 4   |                                 | IDLE (0b00)                | <ul style="list-style-type: none"> <li>CFIFO Mode is modified to disabled mode.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 5   | WAITING FOR TRIGGER (10)        | WAITING FOR TRIGGER (0b10) | <ul style="list-style-type: none"> <li>No trigger occurred.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 6   |                                 | TRIGGERED (0b11)           | <ul style="list-style-type: none"> <li>Appropriate edge or level trigger occurred, OR</li> <li>CFIFO Mode is programmed to single-scan software trigger mode and SSS bit is asserted.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 7   | TRIGGERED (11)                  | IDLE (0b00)                | <ul style="list-style-type: none"> <li>CFIFO in single-scan mode, EQADC detects the EOQ bit asserted at end of command transfer, and CFIFO Mode is not modified to disabled. OR</li> <li>CFIFO, in single-scan level trigger mode, and the gate closes while no commands are being transferred from the CFIFO, and CFIFO Mode is not modified to disabled. OR</li> <li>CFIFO, in single-scan level trigger mode, and EQADC detects a closed gated at end of command transfer, and CFIFO Mode is not modified to disabled. OR</li> <li>CFIFO Mode is modified to disabled mode and CFIFO was not transferring commands.</li> <li>CFIFO Mode is modified to disabled mode while CFIFO was transferring commands, and CFIFO completes or aborts the transfer.</li> </ul>                            |
| 8   |                                 | WAITING FOR TRIGGER (0b10) | <ul style="list-style-type: none"> <li>CFIFO in single or continuous-scan edge trigger mode, EQADC detects the Pause bit asserted at the end of command transfer, the EOQ bit in the same command is negated, and CFIFO Mode is not modified to disabled, OR</li> <li>CFIFO in continuous-scan edge trigger mode, EQADC detects the EOQ bit asserted at the end of command transfer, and CFIFO Mode is not modified to disabled, OR</li> <li>CFIFO, in continuous-scan level trigger mode, and the gate closes while no commands are being transferred from the CFIFO, and CFIFO Mode is not modified to disabled, OR</li> <li>CFIFO, in continuous-scan level trigger mode, and EQADC detects a closed gated at end of command transfer, and CFIFO Mode is not modified to disabled.</li> </ul> |

*Table continues on the next page...*

**Table 29-135. Command FIFO status switching condition  
(continued)**

| No. | From current CFIFO status (CFS) | To new CFIFO status (CFS) | Status switching condition                                                                                               |
|-----|---------------------------------|---------------------------|--------------------------------------------------------------------------------------------------------------------------|
| 9   |                                 | TRIGGERED (0b11)          | <ul style="list-style-type: none"> <li>No event to switch to IDLE or WAITING FOR TRIGGER status has happened.</li> </ul> |

#### 29.6.4.7.2 CQueue completion status

The End of Queue Flag (EOQF) in [EQADC FIFO and Interrupt Status Register \(EQADC\\_FISR<sub>n</sub>\)](#) is asserted when the EQADC completes the transfer of a CFIFO entry with an asserted EOQ bit. Software sets the EOQ bit in the last Command Message of a CQueue to indicate that this entry is the end of the CQueue - see [Message format in EQADC](#) for information on command message formats. The transfer of entries bound for the on-chip ADCs is considered completed when they are stored in the appropriate CBuffer.

The command with a EOQ bit asserted is valid and will be transferred. When EOQIE in [EQADC CFIFO Control Register 0 \(EQADC\\_CFCR0\)](#) and EOQF are asserted, the EQADC will generate an End of Queue interrupt request.

In single-scan modes, command transfers from the corresponding CFIFO will cease when EQADC completes the transfer of a entry with an asserted EOQ. Software involvement is required to rearm the CFIFO so that it can detect new trigger events.

#### Note

An asserted EOQF<sub>x</sub> only implies that EQADC has finished transferring a command with an asserted EOQ bit from CFIFO<sub>x</sub>. It does not imply that result data for the current command and for all previously transferred commands has been returned to the appropriate RFIFO.

#### 29.6.4.7.3 Pause status

In edge trigger mode, when the EQADC completes the transfer of a CFIFO entry with an asserted Pause bit, the EQADC will stop future command transfers from the CFIFO and set the corresponding Pause Flag (PF) in [EQADC FIFO and Interrupt Status Register \(EQADC\\_FISR<sub>n</sub>\)](#). See [Message format in EQADC](#) for information on command message formats. The EQADC ignores the Pause bit in command messages in any software and external level trigger mode. The EQADC sets the PF flag upon detection of

an asserted Pause bit only in single or continuous-scan edge trigger mode. When the PF flag is set for a CFIFO in single-scan edge trigger mode, the SSS bit will not be cleared in [EQADC FIFO and Interrupt Status Register \(EQADC\\_FISR<sub>n</sub>\)](#).

In level trigger mode, the definition of the PF flag has been redefined. In level trigger mode, when CFIFO<sub>x</sub> is in TRIGGERED status, PF<sub>x</sub> is set when CFIFO status changes from TRIGGERED due to detection of a closed gate. The pause flag interrupt routine can be used to verify if the a complete scan of the CQueue was performed. If a closed gate is detected while no command transfers are taking place, it will have immediate effect on the CFIFO status.

When PIE in [EQADC CFIFO Control Register 0 \(EQADC\\_CFCR0\)](#) and PF are asserted, the EQADC will generate a Pause interrupt request.

### Note

In edge trigger mode, an asserted PF<sub>x</sub> only implies that the EQADC finished transferring a command with an asserted PAUSE bit from CFIFO<sub>x</sub>. It does not imply that result data for the current command and for all previously transferred commands has been returned to the appropriate RFIFO.

### Note

In software or level trigger mode, when the EQADC completes the transfer of an entry from CFIFO<sub>x</sub> with an asserted Pause bit, PF<sub>x</sub> will not be set and command transfers will continue without pausing.

#### 29.6.4.7.4 Trigger overrun status

When a CFIFO is configured for edge- or level-trigger mode and is in TRIGGERED state, an additional trigger occurring for the same CFIFO results in a trigger overrun. The trigger overrun bit for the corresponding CFIFO will be set (TORF<sub>x</sub> = 1) in [EQADC FIFO and Interrupt Status Register \(EQADC\\_FISR<sub>n</sub>\)](#). When TORIE in [EQADC CFIFO Control Register 0 \(EQADC\\_CFCR0\)](#) and TORF are asserted, the EQADC generates a trigger overrun interrupt request.

### Note

The trigger overrun flag will not set for CFIFOs configured for software trigger mode.

#### 29.6.4.7.5 Command sequence non-coherency detection

The EQADC provides a mechanism to indicate if a command sequence has been completely executed without interruptions. A command sequence is defined as a group of consecutive commands bound for the same CBuffer and it is expected to be executed without interruptions. A command sequence is coherent if its commands are executed in order without interruptions. Since commands are stored in the CBuffers before being executed in the EQADC, a command sequence is coherent if, while it is transferring commands to a CBuffer, the buffer is only fed with commands from that sequence without ever becoming empty.

A command sequence starts when:

- a CFIFO in TRIGGERED state transfers its first command to CBuffer.
- the CFIFO is constantly transferring commands and the previous command sequence ended.
- the CFIFO resumes command transfers after being interrupted.

And a command sequence ends when:

- an asserted EOQ bit is detected on the last transferred command.
- CFIFO is in edge-trigger mode and asserted PAUSE bit is detected on the last transferred command.
- the CBuffer to which the next command is bound is different from the one to which the last command was transferred.

[Figure 29-133](#) shows examples of how the EQADC would detect command sequences when transferring commands from a CFIFO to a CBuffer. The smallest possible command sequence can have a single command as shown in example 3 of [Figure 29-133](#).



CQueue with a two  
command sequences

|                         |
|-------------------------|
| CF5_CB1_CM0             |
| CF5_CB1_CM1             |
| CF5_CB1_CM2             |
| CF5_CB1_CM3 (Pause = 1) |
| CF5_CB1_CM4             |
| CF5_CB1_CM5             |
| CF5_CB1_CM6 (EOQ = 1)   |

### Example 1

Assuming that these commands are transferred by a CFIFO configured for edge trigger mode and the command transfers are never interrupted, the EQADC would check for non-coherency of two command sequences: one formed by commands 0, 1, 2, 3, and the other by commands 4, 5, 6.

CQueue with a three  
command sequences

|                       |
|-----------------------|
| CF5_CB1_CM0           |
| CF5_CB1_CM1           |
| CF5_CB1_CM2           |
| CF5_CP5_CB0_CM3       |
| CF5_CB0_CM4           |
| CF5_CB1_CM5           |
| CF5_CB1_CM6 (EOQ = 1) |

### Example 2

Assuming that command transfers from the CFIFO are never interrupted, the EQADC would check for non-coherency of three command sequences. The first being formed by commands 0, 1, 2, the second by commands 3, 4 and the third by commands 5, 6. Note that even when the commands of this CQueue are transferred through a CFIFO in continuous-scan mode, the first three commands and the last two commands of this CQueue would still constitute two distinct command sequences, although they are all bound for the same CBuffer, since an asserted EOQ ends a command sequence.

CQueue with a seven  
command sequences

|                       |
|-----------------------|
| CF5_CB1_CM0           |
| CF5_CB2_CM1           |
| CF5_CB3_CM2           |
| CF5_CF5_CB1_CM3       |
| CF5_CB0_CM4           |
| CF5_CB2_CM5           |
| CF5_CB1_CM6 (EOQ = 1) |

### Example 3

The EQADC would check for non-coherency of seven command sequences, all containing a single command, but NCF would never get set.

CF<sub>x</sub>CB<sub>a</sub>CM<sub>n</sub> - Command *n* in CFIFO<sub>x</sub> bound for CBuf

**Figure 29-133. Command sequence examples**

The NCF flag is used to indicate command sequence non-coherency. When the NCF<sub>x</sub> flag is asserted, it indicates that the command sequence being transferred through CFIFO<sub>x</sub> became non-coherent. The NCF flag only becomes asserted for CFIFOs in TRIGGERED state.

A command sequence is non-coherent when, after transferring the first command of a sequence from a CFIFO to a CBuffer, it cannot successively send all the other commands of the sequence before any of the following conditions are true:

- The CFIFO through which commands are being transferred is preempted by a higher priority CFIFO which sends commands to the same CBuffer. The NCF flag becomes asserted immediately after the first command transfer from the preempting CFIFO, that is the higher priority CFIFO, to the CBuffer in use is completed. See [Figure 29-134](#).
- 

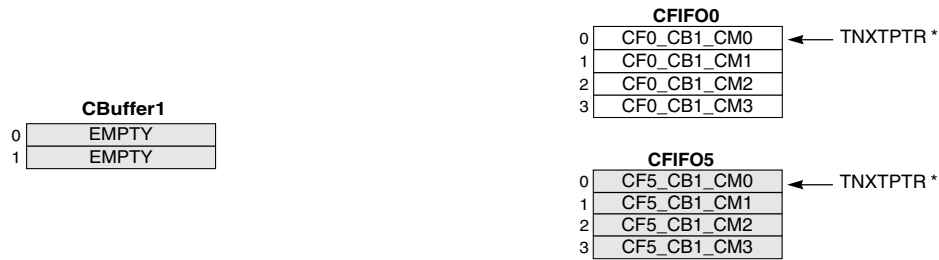
Once a command sequence starts to be transferred, the EQADC will check for the command sequence coherency until the command sequence ends or until one of the conditions below becomes true.

- The command sequence became non-coherent.
- The CFIFO status changed from TRIGGERED.
- The CFIFO underflew.

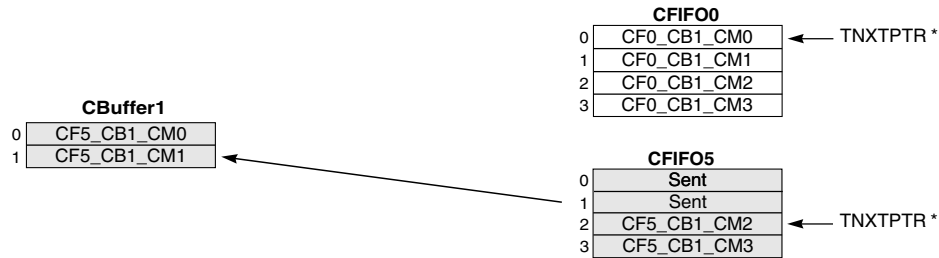
Once command transfers restart/continue, the non-coherency hardware will behave as if the command sequence started from that point. [Figure 29-135](#) depicts how the non-coherency hardware will behave when a non-coherency event is detected.

### Note

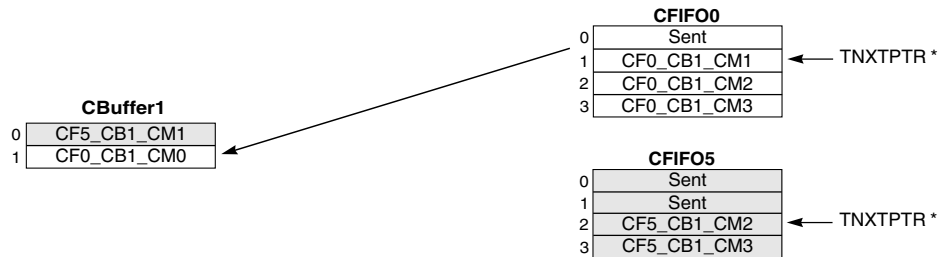
If MODEx is changed to disabled while a CFIFO is transferring commands, the NCF flag for that CFIFO will not become asserted.



(a) CFIFO0 and CFIFO5 both have commands to be sent to CBuffer1, and both are not triggered



(b) CFIFO5 becomes triggered and transfers two commands to CBuffer1



(c) CFIFO0 becomes triggered and transfers a command to CBuffer1. The sequence sent through CFIFO5 becomes non-coherent.

\* TNXTPTR - Transfer Next Data Pointer  
CF<sub>x</sub>\_CB<sub>a</sub>\_CM<sub>n</sub> - Command *n* in CFIFO<sub>x</sub> bound for CBuf

**Figure 29-134. Non-coherency event when different CFIFOs use the same CBuffer**

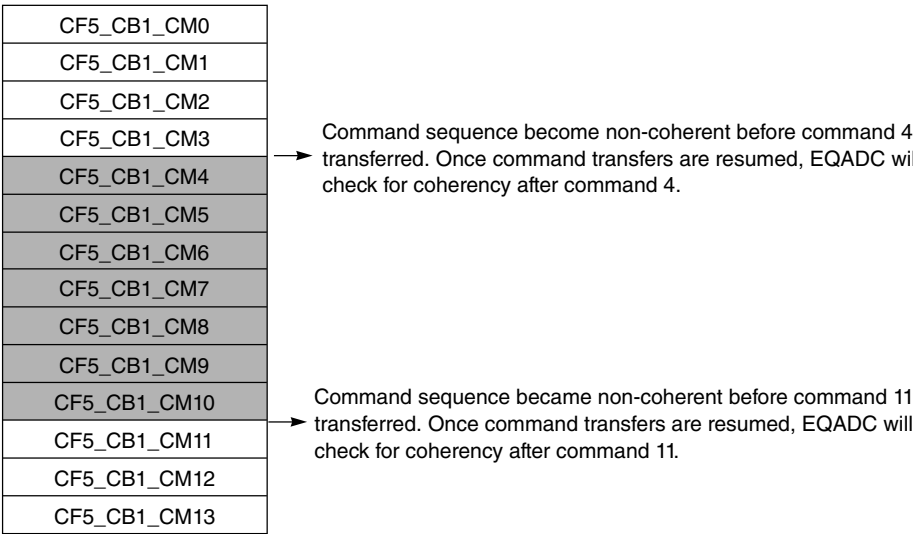


**Figure 29-135. Non-coherency detection when transfers from a command sequence are interrupted**

### 29.6.5 EQADC result FIFOs

#### 29.6.5.1 RFIFO basic functionality

There are six RFIFOs located in the EQADC. Each RFIFO is four entries deep, and each RFIFO entry is 16 bits long. Each RFIFO serves as a temporary storage location for the one of the RQueues allocated in system memory. Result data is saved in the RFIFOs before being moved into the system RQueues. When an RFIFO is not empty, the EQADC sets the corresponding RFDF bit in [EQADC FIFO and Interrupt Status Register \(EQADC\\_FISR<sub>n</sub>\)](#). If RFDE is asserted in [EQADC Interrupt and DMA Control Register 0 \(EQADC\\_IDCR0\)](#), the EQADC generates a request so that an RFIFO entry is moved to the RQueue. An interrupt request, served by the host CPU, is generated when RFDS is negated, and a DMA request, served by the DMAC, is generated when RFDS is asserted. The host CPU or the DMAC responds to these requests by reading [EQADC Result FIFO Pop Register \(EQADC\\_RFPR<sub>n</sub>\)](#) to retrieve data from the RFIFO.

**Note**

The DMAC should be configured to read a single result (16-bit data) from the RFIFO pop registers for every asserted DMA request it acknowledges. See [EQADC/DMAC interface](#) for DMAC configuration guidelines.

**Note**

Reading a word, a half-word, or any bytes from EQADC\_RFPRx will pop an entry from RFIFOx, and the RFCTRx field will be decremented by one.

**Note**

When the DMA is selected ((RFDSx=1), readings should be done only controlled by the DMA protocol. Other ways to flush the RFIFOx to empty the memory can cause a locked DMA request (permanent ON).

Figure 29-136 describes the important components in the RFIFO. Each RFIFO is implemented as a circular set of registers to avoid the need to move all entries at each push/pop operation. The Pop Next Data Pointer always points to the next RFIFO message to be retrieved from the RFIFO when reading EQADC\_RFPR. The Receive Next Data Pointer points to the next available RFIFO location for storing the next incoming message from the on-chip ADCs. The *RFIFO Counter Logic* counts the number of entries in RFIFO and generates interrupt or DMA requests to drain the RFIFO.

POPNXTPTR in [EQADC FIFO and Interrupt Status Register \(EQADC\\_FISR<sub>n</sub>\)](#) indicates which entry is currently being addressed by the Pop Next Data Pointer, and RFCTR, in the same register, provides the number of entries stored in the RFIFO. Using POPNXTPTR and RFCTR, the absolute addresses for Pop Next Data Pointer and Receive Next Data Pointer can be calculated using the following formulas:

Pop Next Data Pointer Address = RFIFOx\_BASE\_ADDRESS + POPNXTPTRx\*4

Receive Next Data Pointer Address = RFIFOx\_BASE\_ADDRESS + [(POPNXTPTRx + RFCTRx) mod RFIFO\_DEPTH] \* 4

where

- $a \bmod b$  returns the remainder of the division of  $a$  by  $b$ .
- RFIFOx\_BASE\_ADDRESS is the smallest memory mapped address allocated to an RFIFOx entry.
- RFIFO\_DEPTH is the number of entries contained in a RFIFO - four in this implementation.

When a new message arrives and RFIFOx is not full, the EQADC copies its contents into the entry pointed by the Receive Next Data Pointer. The RFIFO counter RFCTRx in [EQADC FIFO and Interrupt Status Register \(EQADC\\_FISR<sub>n</sub>\)](#) is incremented by one, and the Receive Next Data Pointer x is also incremented by one (or wrapped around) to

point to the next empty entry in RFIFOx. However, if the RFIFOx is full, the EQADC sets the RFOF in [EQADC FIFO and Interrupt Status Register \(EQADC\\_FISR<sub>n</sub>\)](#). The RFIFOx will not overwrite the older data in the RFIFO, the new data will be ignored, and the Receive Next Data Pointer x is not incremented or wrapped around. RFIFOx is full when the Receive Next Data Pointer x equals the Pop Next Data Pointer x and RFCTR<sub>x</sub> is not zero. RFIFOx is empty when the Receive Next Data Pointer x equals the Pop Next Data Pointer x and RFCTR<sub>x</sub> is zero.

When the EQADC RFIFO Pop Register x is read and the RFIFOx is not empty, the RFIFO counter RFCTR<sub>x</sub> is decremented by one, and the POP Next Data Pointer is incremented by one (or wrapped around) to point to the next RFIFO entry.

When the EQADC RFIFO Pop Register x is read and RFIFOx is empty, EQADC will not decrement the counter value and the POP Next Data Pointer x will not be updated. The read value will be undefined.

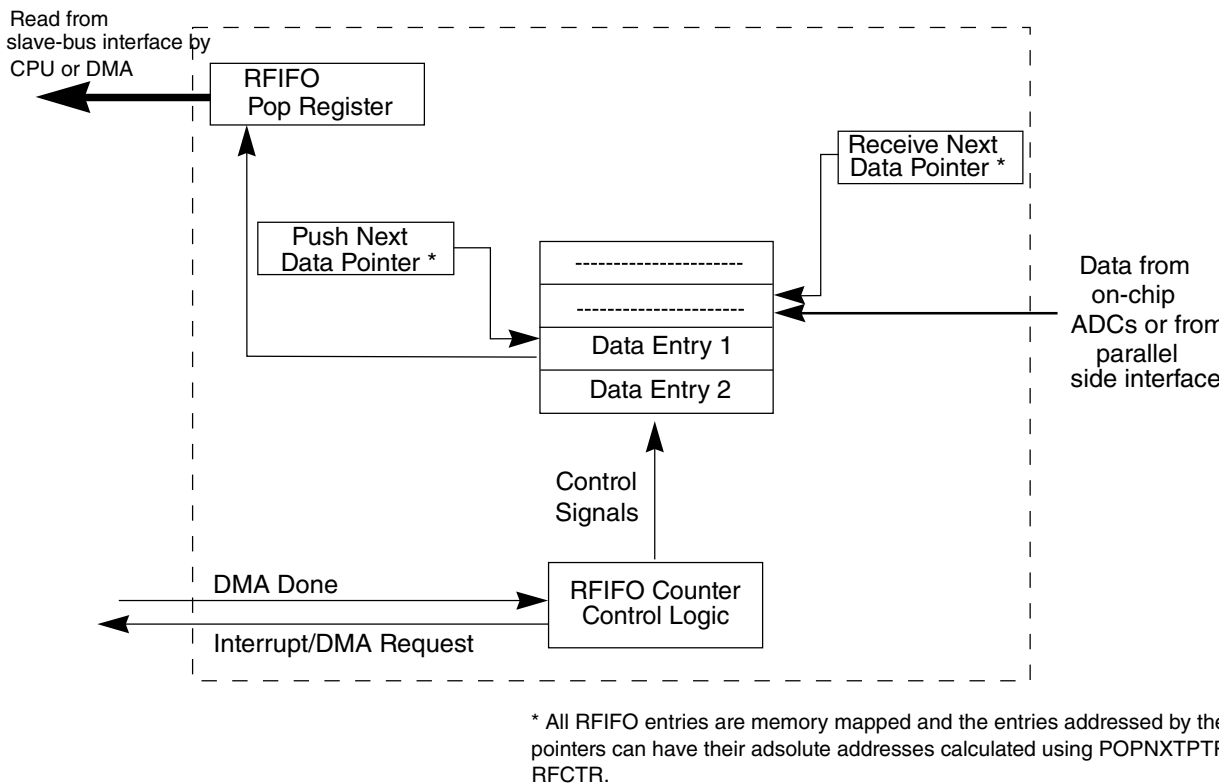
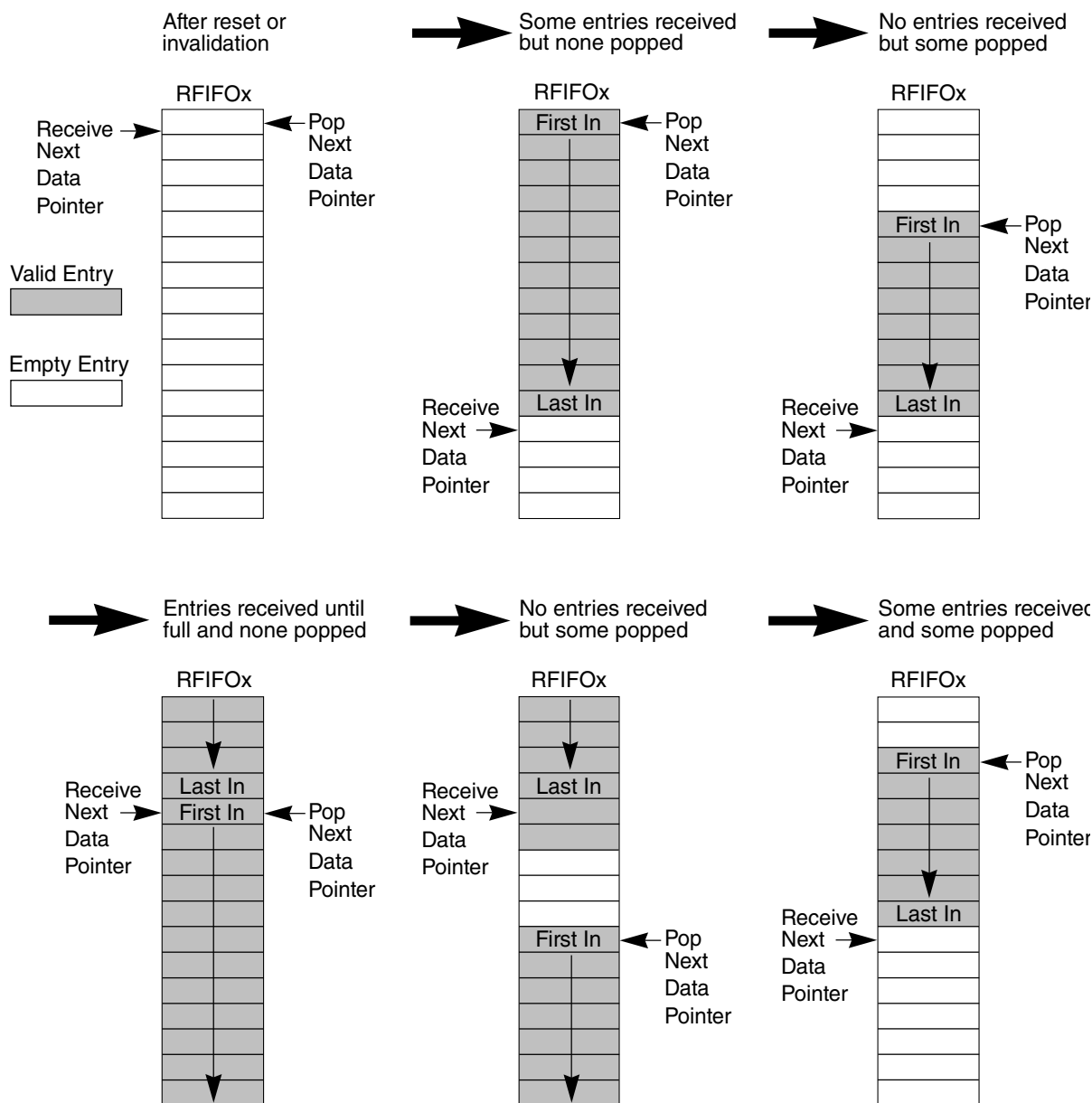


**Figure 29-136. RFIFO diagram**

The detailed behavior of the Pop Next Data Pointer and Receive Next Data Pointer is described in the example shown in [Figure 29-137](#) where an RFIFO with 16 entries is shown for clarity of explanation, the actual hardware implementation has only four entries. In this example, RFIFOx with 16 entries is shown in sequence after popping or receiving entries.



NOTE: x=0, 1, 2, 3, 4,

Figure 29-137. RFIFO entry pointer example

### 29.6.5.2 Distributing result data into RFIFOs

Data to be moved into the RFIFOs can come from different sources: from ADC0, from ADC1, or from the on-chip companion module through the PSI. All result data comes with the MESSAGE\_TAG / MESSAGE\_TAG2 field and the DEST / DEST2 field defining what should be done with the received data. The EQADC hardware decodes the MESSAGE\_TAG / MESSAGE\_TAG2 and DEST/DEST2 fields and:

- stores the 16-bit data into the appropriate RFIFO if the MESSAGE\_TAG / MESSAGE\_TAG2 indicates a valid RFIFO number, or;
- sends the 16-bit data, the MESSAGE\_TAG / MESSAGE\_TAG2 and the DEST / DEST2 data through the PSI to an on-chip companion module (as a decimation filter), or;
- ignores the data in case of a null or "reserved for customer use" MESSAGE\_TAG / MESSAGE\_TAG2.

In general received data is moved into RFIFOs as they become available, while an exception happens when multiple results from different sources become available at the same time. In that case, result data from ADC0 is processed first, result data from ADC1 is only processed after all ADC0 data is processed, and finally returned data from companion module is only processed after all data from ADC0/1 is processed.

When time-stamped results return from the on-chip ADCs, the conversion result and the time stamp are always moved to the RFIFOs in consecutive clock cycles in order to guarantee they are always stored in consecutive RFIFO entries.

## **29.6.6 On-chip ADC configuration and control**

### **29.6.6.1 Enabling and disabling the on-chip ADCs**

The on-chip ADCs have an enable bit (ADC0/1\_EN) in [ADC0/1 Control Registers \(ADC0\\_CR and ADC1\\_CR\)](#) that allows the enabling of the ADCs only when necessary. When the enable bit for an ADC is negated, the clock input to that ADC is stopped. The ADCs are disabled out of reset - ADC0/1\_EN bits are negated - to allow for their safe configuration. The ADC must only be configured when its enable bit is negated. Once the enable bit of an ADC is asserted, clock input to is started.

#### **Note**

Conversion commands sent to the CBuffer of a disabled ADC are ignored by the ADC control hardware.

#### **Note**

Due to legacy reasons, the EQADC will always wait 120 ADC clocks before issuing the first conversion command following the enabling of one of on-chip ADCs, or the exiting of stop mode. There are two independent counters checking for this delay: one clocked by ADC0\_CLK and another by



ADC1\_CLK. Conversion commands can start to be executed whenever one of these counters completes counting 120 ADC clocks.

### 29.6.6.2 ADC clock and conversion speed

The clock input to the ADCs is defined by setting the ADC0/1\_ODD\_PS, the ADC0/1\_CLK\_SEL and the ADC0/1\_CLK\_PS fields in [ADC0/1 Control Registers \(ADC0\\_CR and ADC1\\_CR\)](#). When the ADC0/1\_CLK\_SEL is set, the ADC clock frequency is the same as the system clock, but it has the inverted phase. When it is clear, the ADC0/1\_ODD\_PS and the ADC0/1\_CLK\_PS fields select the clock divide factor by which the system clock will be divided as shown in [Table 29-113](#). The ADC clock frequency is calculated as below and must not exceed ADC\_MAX\_CLOCK. This is also the maximum frequency of system clock when the ADC0/1\_CLK\_SEL is asserted.

$$\text{ADCClockFrequency} = \frac{\text{SystemClockFrequency(MHz)}}{\text{SystemClockDivideFactor}}$$

[Figure 29-138](#) depicts how the ADC clocks for ADC0 and ADC1 are generated.

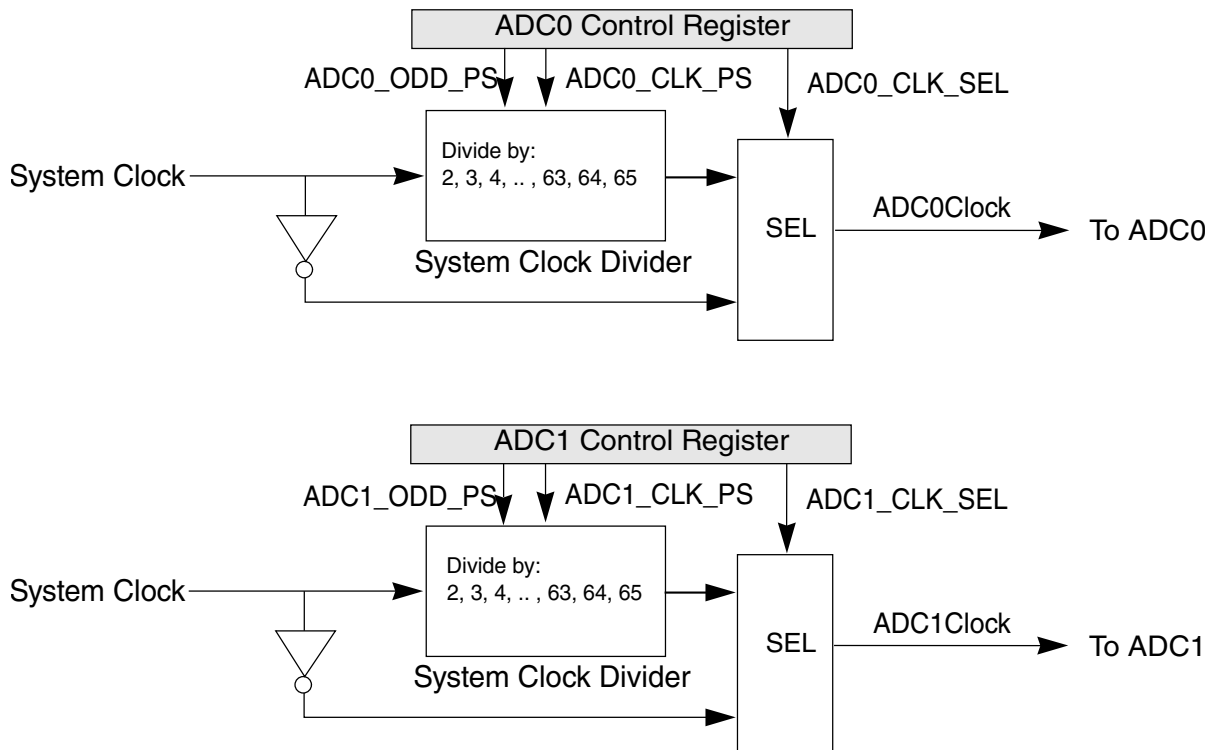


**Figure 29-138. ADC0/1 clock generation**

The ADC conversion speed (in K samples per second: Ksps) is calculated by the following formula. The *number of sampling cycles* is determined by the LST bits in the command message—see [Conversion command format for the standard configuration](#)—

and can take one of the following values: 2, 8, 64, or 128 ADC clock cycles. The *number of AD conversion cycles* is 13 for differential conversions and 14 for single-ended 12-bit resolution and unitary input gain conversions. The maximum conversion speed is achieved when the ADC Clock frequency is set to its maximum and the number of sampling cycles set to its minimum (2 cycles). The maximum conversion speed is ADC\_MAX\_CONV\_RATE.

$$\text{ADCConversionSpeed} = \frac{\text{ADCClockFrequency(MHz)}}{(\text{NumberOfSamplingCycles} + \text{NumberOfADConversionCycles})}$$

Table 29-136 shows an example of how the ADC0/1\_CLK\_PS can be set when using a 120 MHz system clock and the corresponding conversion speeds for all possible ADC clock frequencies. The table also shows that according to the system clock frequency, certain clock divide factors are invalid (2, 4, 6, 8 clock divide factors in the example) because their use would result in a ADC clock frequency higher than the maximum one supported by the ADC. The ADC clock frequency must not exceed ADC\_MAX\_CLOCK.

**Table 29-136. ADC Clock configuration example (system clock frequency=120 MHz)**

| ADC0/1_CLK_PS<br>[0:4] | ADC0/1_ODD_PS | System clock<br>divide factor | ADC clock<br>(system clock =<br>120 MHz) | Differential<br>conversion speed<br>with default sampling<br>time (2 cycles) | Single-ended<br>conversion speed<br>with default sampling<br>time (2 cycles) |
|------------------------|---------------|-------------------------------|------------------------------------------|------------------------------------------------------------------------------|------------------------------------------------------------------------------|
| 0b00000                | 0             | 2                             | N/A                                      | N/A                                                                          | N/A                                                                          |
|                        | 1             | 3                             | N/A                                      | N/A                                                                          | N/A                                                                          |
| 0b00001                | 0             | 4                             | N/A                                      | N/A                                                                          | N/A                                                                          |
|                        | 1             | 5                             | N/A                                      | N/A                                                                          | N/A                                                                          |
| 0b00010                | 0             | 6                             | N/A                                      | N/A                                                                          | N/A                                                                          |
|                        | 1             | 7                             | N/A                                      | N/A                                                                          | N/A                                                                          |
| 0b00011                | 0             | 8                             | 15.0 MHz                                 | 1.0 Msps                                                                     | 938 Ksps                                                                     |
|                        | 1             | 9                             | 13.3 MHz                                 | 889 Ksps                                                                     | 833 Ksps                                                                     |
| 0b00100                | 0             | 10                            | 12.0 MHz                                 | 800 Ksps                                                                     | 750 Ksps                                                                     |
|                        | 1             | 11                            | 10.9 MHz                                 | 727 Ksps                                                                     | 682 Ksps                                                                     |
| 0b00101                | 0             | 12                            | 10.0 MHz                                 | 667 Ksps                                                                     | 625 Ksps                                                                     |
|                        | 1             | 13                            | 9.23 MHz                                 | 615 Ksps                                                                     | 577 Ksps                                                                     |
| 0b00110                | 0             | 14                            | 8.57 MHz                                 | 571 Ksps                                                                     | 536 Ksps                                                                     |
|                        | 1             | 15                            | 8.0 MHz                                  | 533 Ksps                                                                     | 500 Ksps                                                                     |
| 0b00111                | 0             | 16                            | 7.5 MHz                                  | 500 Ksps                                                                     | 469 Ksps                                                                     |
|                        | 1             | 17                            | 7.06 MHz                                 | 471 Ksps                                                                     | 441 Ksps                                                                     |
| 0b01000                | 0             | 18                            | 6.67 MHz                                 | 444 Ksps                                                                     | 417 Ksps                                                                     |
|                        | 1             | 19                            | 6.32 MHz                                 | 421 Ksps                                                                     | 395 Ksps                                                                     |
| 0b01001                | 0             | 20                            | 6.0 MHz                                  | 400 Ksps                                                                     | 375 Ksps                                                                     |
|                        | 1             | 21                            | 5.71 MHz                                 | 381 Ksps                                                                     | 357 Ksps                                                                     |

Table continues on the next page...

**Table 29-136. ADC Clock configuration example (system clock frequency=120 MHz)  
(continued)**

| ADC0/1_CLK_PS<br>[0:4] | ADC0/1_ODD_PS | System clock<br>divide factor | ADC clock<br>(system clock =<br>120 MHz) | Differential<br>conversion speed<br>with default sampling<br>time (2 cycles) | Single-ended<br>conversion speed<br>with default sampling<br>time (2 cycles) |
|------------------------|---------------|-------------------------------|------------------------------------------|------------------------------------------------------------------------------|------------------------------------------------------------------------------|
| 0b01010                | 0             | 22                            | 5.45 MHz                                 | 364 Ksps                                                                     | 341 Ksps                                                                     |
|                        | 1             | 23                            | 5.22 MHz                                 | 348 Ksps                                                                     | 326 Ksps                                                                     |
| 0b01011                | 0             | 24                            | 5.0 MHz                                  | 333 Ksps                                                                     | 313 Ksps                                                                     |
|                        | 1             | 25                            | 4.80 MHz                                 | 320 Ksps                                                                     | 300 Ksps                                                                     |
| 0b01100                | 0             | 26                            | 4.62 MHz                                 | 308 Ksps                                                                     | 288 Ksps                                                                     |
|                        | 1             | 27                            | 4.44 MHz                                 | 296 Ksps                                                                     | 278 Ksps                                                                     |
| 0b01101                | 0             | 28                            | 4.29 MHz                                 | 286 Ksps                                                                     | 268 Ksps                                                                     |
|                        | 1             | 29                            | 4.14 MHz                                 | 276 Ksps                                                                     | 259 Ksps                                                                     |
| 0b01110                | 0             | 30                            | 4.0 MHz                                  | 267 Ksps                                                                     | 250 Ksps                                                                     |
|                        | 1             | 31                            | 3.87 MHz                                 | 258 Ksps                                                                     | 242 Ksps                                                                     |
| 0b01111                | 0             | 32                            | 3.75 MHz                                 | 250 Ksps                                                                     | 234 Ksps                                                                     |
|                        | 1             | 33                            | 3.64 MHz                                 | 242 Ksps                                                                     | 227 Ksps                                                                     |
| 0b10000                | 0             | 34                            | 3.53 MHz                                 | 235 Ksps                                                                     | 221 Ksps                                                                     |
|                        | 1             | 35                            | 3.43 MHz                                 | 229 Ksps                                                                     | 214 Ksps                                                                     |
| 0b10001                | 0             | 36                            | 3.33 MHz                                 | 222 Ksps                                                                     | 208 Ksps                                                                     |
|                        | 1             | 37                            | 3.24 MHz                                 | 216 Ksps                                                                     | 203 Ksps                                                                     |
| 0b10010                | 0             | 38                            | 3.16 MHz                                 | 211 Ksps                                                                     | 198 Ksps                                                                     |
|                        | 1             | 39                            | 3.08 MHz                                 | 205 Ksps                                                                     | 192 Ksps                                                                     |
| 0b10011                | 0             | 40                            | 3.0 MHz                                  | 200 Ksps                                                                     | 188 Ksps                                                                     |
|                        | 1             | 41                            | 2.93 MHz                                 | 195 Ksps                                                                     | 183 Ksps                                                                     |
| 0b10100                | 0             | 42                            | 2.86 MHz                                 | 190 Ksps                                                                     | 179 Ksps                                                                     |
|                        | 1             | 43                            | 2.79 MHz                                 | 186 Ksps                                                                     | 174 Ksps                                                                     |
| 0b10101                | 0             | 44                            | 2.73 MHz                                 | 182 Ksps                                                                     | 170 Ksps                                                                     |
|                        | 1             | 45                            | 2.67 MHz                                 | 178 Ksps                                                                     | 167 Ksps                                                                     |
| 0b10110                | 0             | 46                            | 2.61 MHz                                 | 174 Ksps                                                                     | 163 Ksps                                                                     |
|                        | 1             | 47                            | 2.55 MHz                                 | 170 Ksps                                                                     | 160 Ksps                                                                     |
| 0b10111                | 0             | 48                            | 2.5 MHz                                  | 167 Ksps                                                                     | 156 Ksps                                                                     |
|                        | 1             | 49                            | 2.45 MHz                                 | 163 Ksps                                                                     | 153 Ksps                                                                     |
| 0b11000                | 0             | 50                            | 2.4 MHz                                  | 160 Ksps                                                                     | 150 Ksps                                                                     |
|                        | 1             | 51                            | 2.35 MHz                                 | 157 Ksps                                                                     | 147 Ksps                                                                     |
| 0b11001                | 0             | 52                            | 2.31 MHz                                 | 154 Ksps                                                                     | 144 Ksps                                                                     |
|                        | 1             | 53                            | 2.26 MHz                                 | 151 Ksps                                                                     | 142 Ksps                                                                     |
| 0b11010                | 0             | 54                            | 2.22 MHz                                 | 148 Ksps                                                                     | 139 Ksps                                                                     |
|                        | 1             | 55                            | 2.18 MHz                                 | 145 Ksps                                                                     | 136 Ksps                                                                     |
| 0b11011                | 0             | 56                            | 2.14 MHz                                 | 143 Ksps                                                                     | 134 Ksps                                                                     |
|                        | 1             | 57                            | 2.11 MHz                                 | 140 Ksps                                                                     | 132 Ksps                                                                     |

*Table continues on the next page...*

**Table 29-136. ADC Clock configuration example (system clock frequency=120 MHz)  
(continued)**

| ADC0/1_CLK_PS<br>[0:4] | ADC0/1_ODD_PS | System clock<br>divide factor | ADC clock<br>(system clock =<br>120 MHz) | Differential<br>conversion speed<br>with default sampling<br>time (2 cycles) | Single-ended<br>conversion speed<br>with default sampling<br>time (2 cycles) |
|------------------------|---------------|-------------------------------|------------------------------------------|------------------------------------------------------------------------------|------------------------------------------------------------------------------|
| 0b11100                | 0             | 58                            | 2.07 MHz                                 | 138 Ksps                                                                     | 129 Ksps                                                                     |
|                        | 1             | 59                            | 2.03 MHz                                 | 136 Ksps                                                                     | 127 Ksps                                                                     |
| 0b11101                | 0             | 60                            | 2.0 MHz                                  | 133 Ksps                                                                     | 125 Ksps                                                                     |
|                        | 1             | 61                            | 1.97 MHz                                 | 131 Ksps                                                                     | 123 Ksps                                                                     |
| 0b11110                | 0             | 62                            | 1.94 MHz                                 | 129 Ksps                                                                     | 121 Ksps                                                                     |
|                        | 1             | 63                            | 1.90 MHz                                 | 127 Ksps                                                                     | 119 Ksps                                                                     |
| 0b11111                | 0             | 64                            | 1.88 MHz                                 | 125 Ksps                                                                     | 117 Ksps                                                                     |
|                        | 1             | 65                            | 1.85 MHz                                 | 123 Ksps                                                                     | 115 Ksps                                                                     |

### 29.6.6.3 Time stamp feature

The on-chip ADCs can provide a time stamp for the conversions they execute. A time stamp is the value of the time base counter latched when the EQADC detects the end of the analog input voltage sampling. A time stamp for a conversion command is requested by setting the TSR bit in the corresponding command. When TSR is negated, that is a time stamp is not requested, the ADC returns a single result message containing the conversion result. When TSR is asserted, that is a time stamp is requested, the ADC returns two result messages; one containing the conversion result, and afterwards another containing the time stamp for that conversion. The result messages are sent in this order to the RFIFOs and both messages are sent to the same RFIFO was specified in the MESSAGE\_TAG field of the executed conversion command.

The time stamp can be provided by an external source using the STAC bus (more details in [STAC bus client sub-module \(STACC\)](#)) or by the internal time base counter. The selection between the two sources is done by field ADC0/1\_TBSEL in the ADC0/1\_CR register or by field ATBSEL in registers ADC\_ACR1-14. See [Table 29-112](#) and the ATBSEL field description in [Alternate Configuration 1-14 Control Registers \(ADC\\_ACR1-14\)](#) for selection details.

The time base counter is a 16-bit up counter that wraps after reaching 0xFFFF. It is disabled after reset and it is enabled according to the setting of TBC\_CLK\_PS field in [ADC Time Stamp Control Register \(ADC\\_TSCR\)](#). TBC\_CLK\_PS defines if the counter is enabled or disabled, and, if enabled, at what frequency it is incremented. The time stamps are returned regardless of whether the time base counter is enabled or disabled. The time base counter can be reset by writing 0x0000 to [ADC Time Base Counter Registers \(ADC\\_TBCR\)](#) with a write configuration command.

### 29.6.6.3.1 STAC bus client sub-module (STACC)

The STAC sub-block STACC extracts the external time base from the STAC bus to the EQADC. Figure 29-139 provides a block diagram for the STACC sub-module that contains actually two slot selectors.

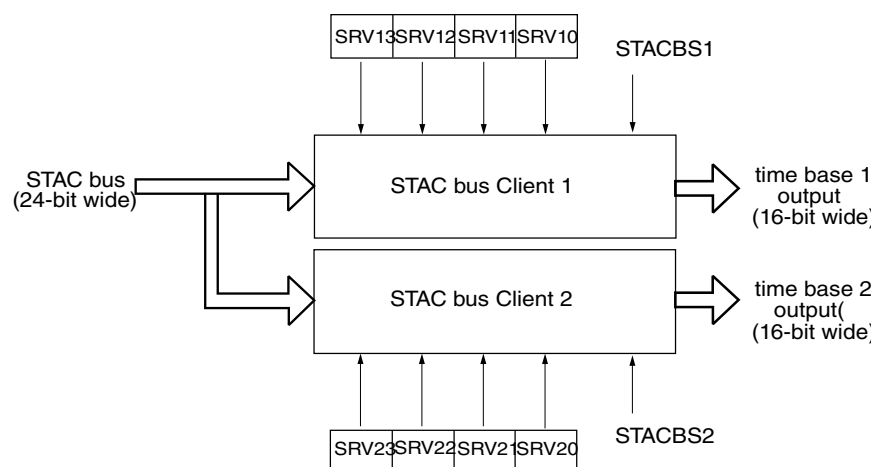
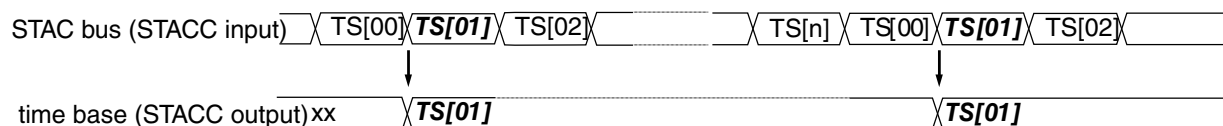


**Figure 29-139. STAC bus client block diagram**

The STAC bus data stream is composed by several identified time slots. Each time slot contains a time value from a specified source. The bits SRV1/2[0:3] in register EQADC\_STACCR are used to select the desired time slots of the STAC bus to be used internally by the EQADC. Figure 29-140 shows a timing diagram for the STACC.



- NOTES: 1. Maximum of 16 Time Slots (TS[n])  
2. In this case, SRV bits were set to capture TS[01]

**Figure 29-140. Timing diagram for the STAC bus and STACC output**

Every time the selected time slot change, the STACC outputs are updated. As this time base is an external data to the EQADC, this output is not affected by the stop or by the debug internal state.

After the slot selection is done and the timebase data is extracted, the STACBS1/2 bits select 16 bits from the original 24-bit timebase data. These selected bits are the timebase to be used internal to the EQADC.

### 29.6.6.4 ADC pre-gain feature

Each ADC can be configured to have a selectable input gain as defined in [Alternate Configuration 1-14 Control Registers \(ADC\\_ACR1-14\)](#). This means the input signal is sampled and the result is amplified by factor 2, or 4 before the conversion phase. In present implementation of this feature, the conversion is 1 or 2 ADC clock cycles longer for gain 2 or gain 4, respectively.

### 29.6.6.5 ADC resolution selection feature

The ADCs conversion resolutions can be 8 bits, 10 bits or 12 bits as described in [Alternate Configuration 1-14 Control Registers \(ADC\\_ACR1-14\)](#). For conversions at a resolution less than 12, the ADC is executing less operations and the conversion time is smaller. In this ADC, it is verified that there is 1 ADC clock cycle for each bit of resolution. Therefore, for the same ADC clock frequency, the ADC sample frequency is higher for lower resolutions.

When a conversion is undertaken at a resolution less than 12, the result is presented by the ADC in right justified format in the 12-bit input bus e.g.: 0000xxxxxxx for 8 bits and 00xxxxxxxxxx for 10 bits. The EQADC inverts the result to left justified format i.e.: xxxxxxxx0000 for 8 bits and xxxxxxxxxx00 for 10 bits. This is because the same calibration coefficients in the MAC can then be used. The left shift operation is done just after the conversion result enters the EQADC, in the Resolution Adjustment block prior to the MAC, as illustrated in [Figure 29-143](#).

### 29.6.6.6 ADC calibration feature

#### 29.6.6.6.1 Overview

There are three sets of calibration coefficients for each ADC. Each set is composed by a gain factor and an offset factor: GCCn/OCCn, ALTGCCn1/ALTGCCn1, and ALTGCCn2/ALTGCCn2, where n is the ADC number 0 or 1. The pair GCCn/OCCn is selected when it is used the normal configuration or the alternate configurations 3 to 8. The pair ALTGCCn1/ALTGCCn1 is used only when the alternate configuration 1 is selected. And the pair ALTGCCn2/ALTGCCn2 is for the alternate configuration 2. The description below is for a generic pair of gain/offset GCC/OCC.

The EQADC provides a calibration scheme to remove the effects of gain and offset errors from the results generated by the on-chip ADCs. Only results generated by the on-chip ADCs are calibrated. The main component of calibration hardware is a Multiply-and-Accumulate (MAC) unit, one per on-chip ADC, that is used to calculate the following transfer function which relates a calibrated result to a raw, uncalibrated one.

$$\text{CAL\_RES} = \text{GCC} * \text{RAW\_RES} + \text{OCC} + 2;$$

where:

- CAL\_RES is the calibrated result corresponding the input voltage  $V_i$ .
- GCC is the gain calibration constant.
- RAW\_RES is the raw, uncalibrated result with resolution adjustment corresponding to an specific input voltage  $V_i$ .
- OCC is the offset calibration constant.
- The addition of two reduces the maximum quantization error of the ADC. See [Quantization error reduction during calibration](#).

Calibration constants GCC and OCC are determined by taking two samples of known reference voltages and using these samples to calculate the values for the constants. For details and an example about how to calculate the calibration constants and use them in result calibration see [ADC Result Calibration](#). Once calculated, GCC is stored in [ADC0/1 Gain Calibration Constant Registers \(ADC0\\_GCCR and ADC1\\_GCCR\)](#) and OCC in [ADC0/1 Offset Calibration Constant Registers \(ADC0\\_OCCR and ADC1\\_OCCR\)](#), from where their values are fed to the MAC unit. The alternate gain values are stored in [ADC0/1 Alternate Gain Registers \(ADC0\\_AGR1-2 and ADC1\\_AGR1-2\)](#) and the alternate offset values in [ADC0/1 Alternate Offset Register \(ADC0\\_AOR1-2 and ADC1\\_AOR1-2\)](#). Since the analog characteristics of each on-chip ADCs differs, each ADC has an independent pair of calibration constants.

A conversion result is calibrated according to the status of CAL bit in the command that initiated the conversion. If the CAL bit is asserted, the EQADC will automatically calculate the calibrated result before sending the result to the appropriate RFIFO or companion module. If the CAL bit is negated, the result is not calibrated, it bypasses the calibration hardware, and is directly sent to the appropriate RFIFO or companion module.

#### 29.6.6.6.2 MAC unit and operand data format

The MAC unit diagram is shown in [Figure 29-141](#). Each on-chip ADC has a separate MAC unit to fine-tune its conversion results. The description below considers the general calibration constant registers but it is the same for the alternate calibration constants.

The OCC0/1 operand is a 14-bit signed value stored in [ADC0/1 Offset Calibration Constant Registers \(ADC0\\_OCCR and ADC1\\_OCCR\)](#)." The RAW\_RES operand is the raw uncalibrated result, and it is the direct output from the on-chip ADCs but passing through the resolution adjustment block. The GCC0/1 operand is a 15-bit fixed point unsigned value stored in [ADC0/1 Gain Calibration Constant Registers \(ADC0\\_GCCR and ADC1\\_GCCR\)](#)." The GCC is expressed in the *GCC\_INT.GCC\_FRAC* binary format. The integer part of the GCC (GCC\_INT=GCC[1]) contains a single binary digit while its fractional part (GCC\_FRAC=GCC[2:15]) contains 14 bits - see [Figure 29-142](#). The gain constant equivalent decimal value ranges from 0 to 1.999938..., as shown in [Table 29-138](#). Two is always added to the MAC output - see [Quantization error reduction during calibration](#). CAL\_RES output is the calibrated result, and it is a 14-bit unsigned value. CAL\_RES is truncated to 0x3FFF, in case of a overflow, and to 0x0000, in case of an underflow.

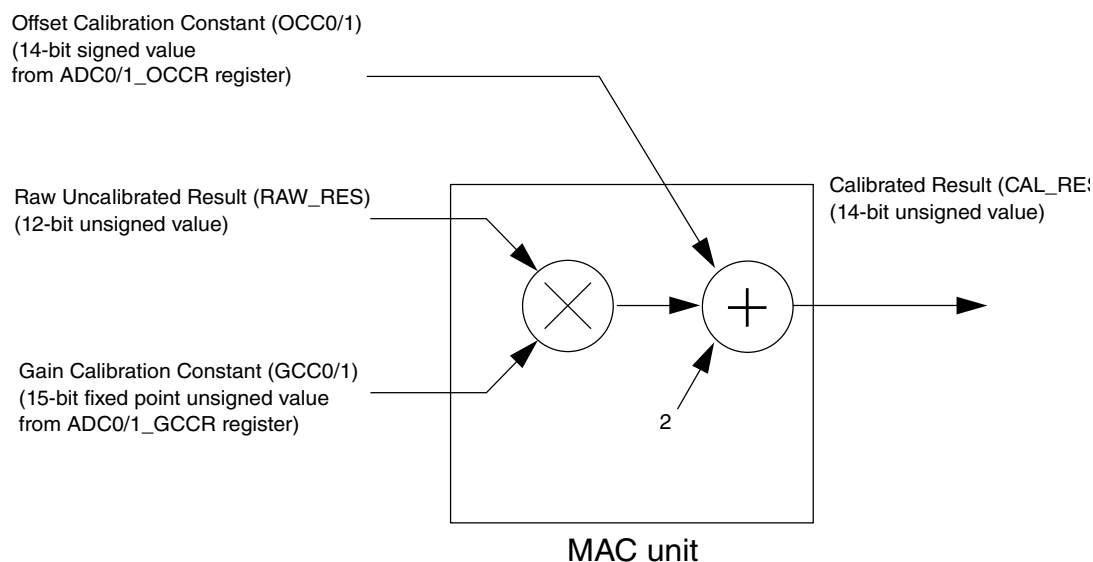


Figure 29-141. MAC unit diagram



Figure 29-142. Gain calibration constant format

Table 29-137. Gain calibration constant format — field descriptions

| Field          | Description                                                                                                                      |
|----------------|----------------------------------------------------------------------------------------------------------------------------------|
| GCC_INT        | Integer part of the gain calibration constant for ADC0/1GCC_INT is the integer part of the gain calibration constant for ADC0/1. |
| GCC_FRAC[1:14] | Fractional part of the gain calibration constant for ADC0/1. See <a href="#">Table 29-138</a> .                                  |



**Table 29-137. Gain calibration constant format — field descriptions**

| Field | Description                                                                                                                                    |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------|
|       | GCC_FRAC is the fractional part of the gain calibration constant for ADC0/1. GCC_FRAC can express decimal values ranging from 0 to 0.999938... |

**Table 29-138. Binary and decimal representations of the gain constant**

| Gain constant<br>(GCC_INT.GCC_FRAC binary format) | Corresponding decimal value |
|---------------------------------------------------|-----------------------------|
| 0.0000_0000_0000_00                               | 0                           |
| ...                                               | ...                         |
| 0.1000_0000_0000_00                               | 0.5                         |
| ...                                               | ...                         |
| 0.1111_1111_1111_11                               | 0.999938...                 |
| 1.0000_0000_0000_00                               | 1                           |
| ...                                               | ...                         |
| 1.1100_0000_0000_00                               | 1.75                        |
| ...                                               | ...                         |
| 1.1111_1111_1111_11                               | 1.999938...                 |

### 29.6.6.7 ADC control logic overview and command execution

Figure 29-143 shows the basic logic blocks involved in the ADC Control and how they interact. CFIFOs/RFIFOs interact with CBuffers/*Abort Cont/Result Message Return Logic* through the *FIFO Control Unit*. The EB and BN bits in the Command Message uniquely identify the CBuffer to which a command should be sent. The *FIFO Control Unit* decodes these bits and sends the ADC command to the proper CBuffer. Other blocks of logic are the *Resolution Adjustment*, *Result Format and Calibration Sub-Block*, the *Time Stamp Logic*, and the *MUX Control Logic*.

The *Resolution Adjustment Sub-Block* receives the 12-bit data bus directly from the ADC and changes the received conversion results from right aligned format of ADC to the left aligned format depending on the selected resolution of the conversion. This operation helps the calibration processing to use the calibration coefficients always with the same format.

The *Result Format and Calibration Sub-Block* formats the returning data into Result Messages and sends them to the RFIFOs <sup>6</sup>. The returning data can be data read from an ADC register, a conversion result, or a time stamp. The formatting and calibration of conversion results also take place inside this sub-block.

The *Time Stamp Logic* latches the value of the time base counter or the STAC bus time base when detecting the end of the analog input voltage sampling, and sends it to the *Result Format and Calibration Sub-Block* as time stamp information.

The *MUX Control Logic* generates the proper MUX control signals and, when the ADC0/1\_EMUX bits are asserted, the MA signals based on the channel numbers extracted from the ADC Command.

When the on-chip ADC abort feature is not enabled, ADC Commands are stored in the CBuffers as they come and they are executed in the first-in-first-out basis. After the execution of a command in ENTRY1 finishes all commands are shifted one entry. After the shift, ENTRY0 is always empty and ready to receive a new command. Execution of configuration commands only start when they reach ENTRY1. Consecutive conversion commands are pipelined and their execution can start while in ENTRY0. This is explained below.

AD conversion accuracy can be affected by the settling time of the input channel multiplexers. Some time is required for the channel multiplexers internal capacitances to settle after the channel number is changed. If the time prior to sampling is not long enough to absorb this settling, then the settling time will take from ADC sampling time which may result in inaccurate sampling and ultimately compromise conversion result accuracy - see [Figure 29-144](#) (a). This could be avoided by switching the multiplexers in preparation for the next command's sampling during the AD conversion phase of the current command as showed in [Figure 29-144](#) (b). In EQADC, this is done in the following way; when a conversion command is in buffer ENTRY1 and another conversion command is identified in ENTRY0, then the channel number of ENTRY0 is sent to the *MUX Control Logic* some cycles before the sampling phase of the command in ENTRY0 starts. In this way, sampling for the next command can promptly start after the current conversion finishes because the internal capacitance of the multiplexers will be settled by that time, allowing for more accurate sampling. This is specially important for applications that require high conversion speeds, that is with the ADC running at maximum clock frequency and with the analog input voltage sampling time set to a minimum (2 ADC clock cycles), when the short sampling time does not allow the multiplexers to completely settle. The second advantage of pipelining conversion

6. The result messages may also be routed to an on-chip companion module via the side interface, and then fed back to the RFIFOs.

commands is to provide precise conversion intervals, which means the time intervals between two consecutive conversions are the same. This is important for any digital signal process application.

When the on-chip ADC abort feature is enabled, ADC Commands from CFIFO0 should be considered immediately, even stopping the execution of some command that is already in ENTRY1. When the abort request is sent to the ADC, the already stored commands in the CBuffers are copied in a temporary set of registers. The first ADC command from CFIFO0 is sent after the abort acknowledge indication from ADC. The process is the same as usual until the transfer of the last command from CFIFO0. Then the temporarily stored commands that were postponed by the abortion are recovered and they are pipelined for execution. After the last command from this temporary memory is transferred, the next commands are pipelined from the CFIFOs.

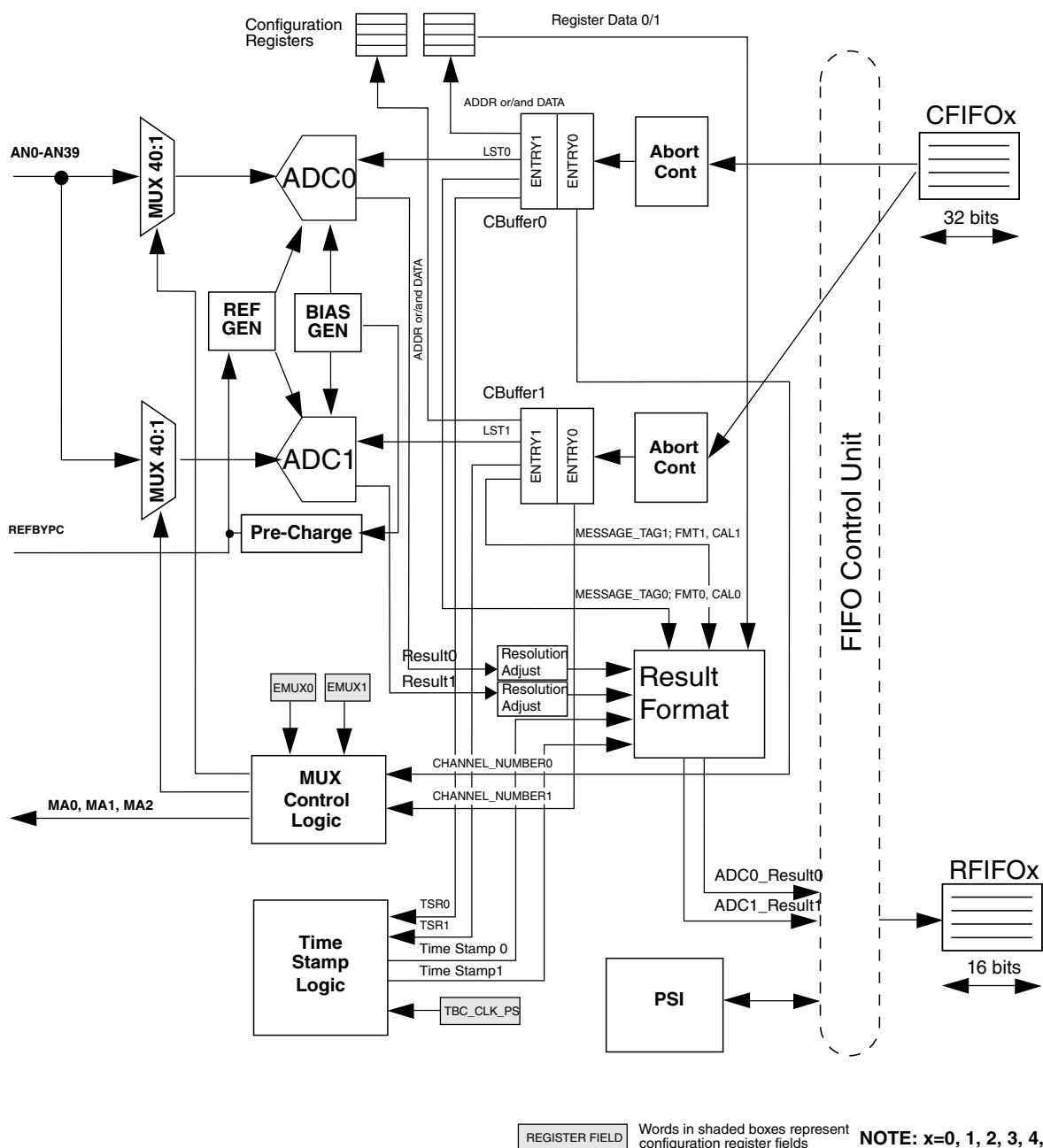
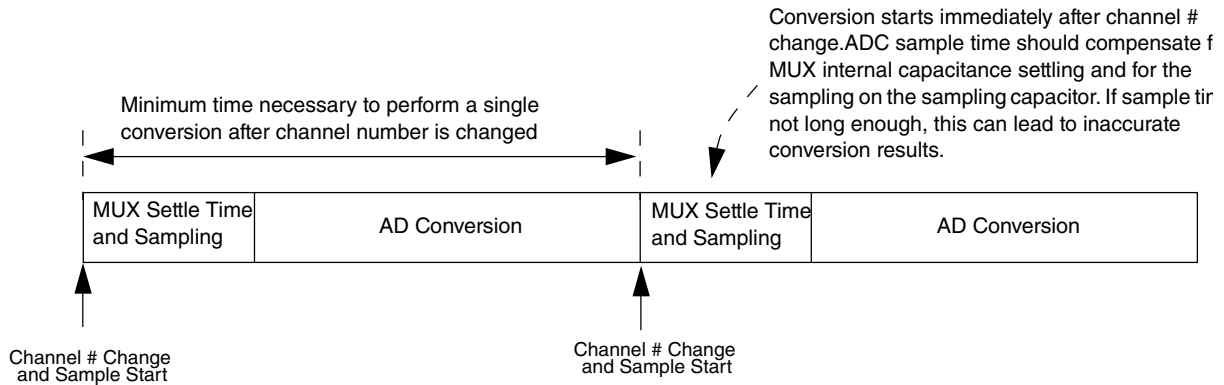
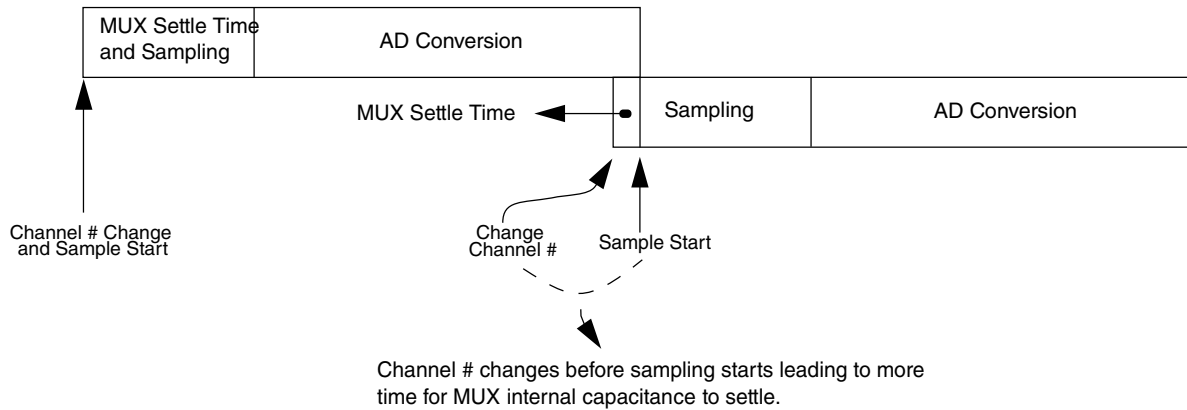


Figure 29-143. On-chip ADC control scheme



(a) Command Execution Sequence for Two Non-Overlapped Commands



(b) Command Execution Sequence for Two Overlapped Commands

**Figure 29-144. Overlapping consecutive conversion commands**

## 29.6.7 Internal/external multiplexing

### 29.6.7.1 Channel assignment

The internal analog multiplexers select one of the 40 analog input pins for conversion, based on the CHANNEL\_NUMBER field of a Command Message. The analog input pin channel number assignments and the pin definitions vary depending on how the ADC0/1\_EMUX are configured. Allowed combinations of ADC0/1\_EMUX bits are shown in [Table 29-139](#) together with references to tables indicating how CHANNEL\_NUMBER field of each conversion command must be set to avoid channel selection conflicts.

During differential conversions the analog multiplexer passes differential signals to both the positive and negative terminals of the ADC. The differential conversions can only be initiated on four channels: DAN0, DAN1, DAN2, and DAN3. See chip-specific section for the channel numbers used to select differential conversions.

**Table 29-139. ADC0/1\_EMUX bits combinations**

| ADC0_EMUX | ADC1_EMUX | CHANNEL_NUMBER should be set as in    |                                       |
|-----------|-----------|---------------------------------------|---------------------------------------|
|           |           | ADC0                                  | ADC1                                  |
| 0         | 0         | See chip-specific section for details | See chip-specific section for details |
| 0         | 1         | See chip-specific section for details | See chip-specific section for details |
| 1         | 0         | See chip-specific section for details | See chip-specific section for details |
| 1         | 1         | Reserved <sup>1</sup>                 |                                       |

1. ADC0\_EMUX and ADC1\_EMUX must not be asserted at the same time

See the chip-specific EQADC information for channel number assignments for the non-multiplexed mode. The two on-chip ADCs can access the same analog input pins but simultaneous conversions are not allowed. Also, when one ADC is performing a differential conversion on a pair of pins, the other ADC must not access either of these two pins as single-ended channels. The 43 single-ended channels and 4 differential pairs are shared between the two ADCs.

See the chip-specific EQADC information for the channel number assignments for multiplexed mode. The two on-chip ADCs can access the same analog input pins but simultaneous conversions are not allowed. Also, when one ADC is performing a differential conversion on a pair of pins, the other ADC must not access either of these two pins as single-ended channels. The ADC with the ADC0/1\_EMUX bit asserted can access 4 differential pairs, 39 single-ended, and, at most, 64 externally multiplexed channels. See [External multiplexing](#) for a detailed explanation about how external multiplexing can be achieved.

### 29.6.7.2 External multiplexing

The EQADC can use from one to eight external multiplexer chips to expand the number of analog signals that may be converted. Up to 64 analog channels can be converted through external multiplexer selection. The externally multiplexed channels are automatically selected by the CHANNEL\_NUMBER field of a Command Message, in the same way done with internally multiplexed channels. The software selects the external multiplexed mode by setting the ADC0/1\_EMUX bit in either ADC0\_CR or ADC1\_CR depending on which ADC will perform the conversion. See the chip-specific EQADC information for the channel number assignments for the multiplexed mode.

There are 4 differential pairs, 39 single-ended, and, at most, 64 externally multiplexed channels which can be selected. Only one ADC can have its ADC0/1\_EMUX bit asserted at a time.

Figure 29-145 shows the maximum configuration of eight external multiplexer chips connected to the EQADC. The external multiplexer chip selects one of eight analog inputs and connects it to a single analog output, which is fed to a specific input of the EQADC. The EQADC provides three multiplexed address signals, MA0, MA1, and MA2, to select one of eight inputs. These three multiplexed address signals are connected to all eight external multiplexer chips. The analog output of the eight multiplex chips are each connected to eight separate EQADC inputs, ANR, ANS, ANT, ANU, ANW, ANX, ANY, and ANZ. The MA pins correspond to the three least significant bits of the channel number that selects ANR, ANS, ANT, ANU, ANW, ANX, ANY, and ANZ with MA0 being the most significant bit - See Table 29-140. In that table, '0' means pin is driven LOW and '1' that pin is driven HIGH.

**Table 29-140. Encoding of MA pins**

| Channel number selecting ANR, ANS, ANT, ANU, ANW, ANX, ANY, ANZ (decimal) |     |     |     |     |     |     |     | MA0 | MA1 | MA2 |
|---------------------------------------------------------------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| ANR                                                                       | ANS | ANT | ANU | ANW | ANX | ANY | ANZ |     |     |     |
| 224                                                                       | 232 | 240 | 248 | 64  | 72  | 80  | 88  | 0   | 0   | 0   |
| 225                                                                       | 233 | 241 | 249 | 65  | 73  | 81  | 89  | 0   | 0   | 1   |
| 226                                                                       | 234 | 242 | 250 | 66  | 74  | 82  | 90  | 0   | 1   | 0   |
| 227                                                                       | 235 | 243 | 251 | 67  | 75  | 83  | 91  | 0   | 1   | 1   |
| 228                                                                       | 236 | 244 | 252 | 68  | 76  | 84  | 92  | 1   | 0   | 0   |
| 229                                                                       | 237 | 245 | 253 | 69  | 77  | 85  | 93  | 1   | 0   | 1   |
| 230                                                                       | 238 | 246 | 254 | 70  | 78  | 86  | 94  | 1   | 1   | 0   |
| 231                                                                       | 239 | 247 | 255 | 71  | 79  | 87  | 95  | 1   | 1   | 1   |

When the external multiplexed mode is selected for either ADC, the EQADC automatically creates the MA output signals from CHANNEL\_NUMBER field of a Command Message. The EQADC also converts the proper input channel (ANR, ANS, ANT, ANU, ANW, ANX, ANY, and ANZ) by interpreting the CHANNEL\_NUMBER field. As a result, up to 64 externally multiplexed channels appear to the conversion queues as directly connected signals.

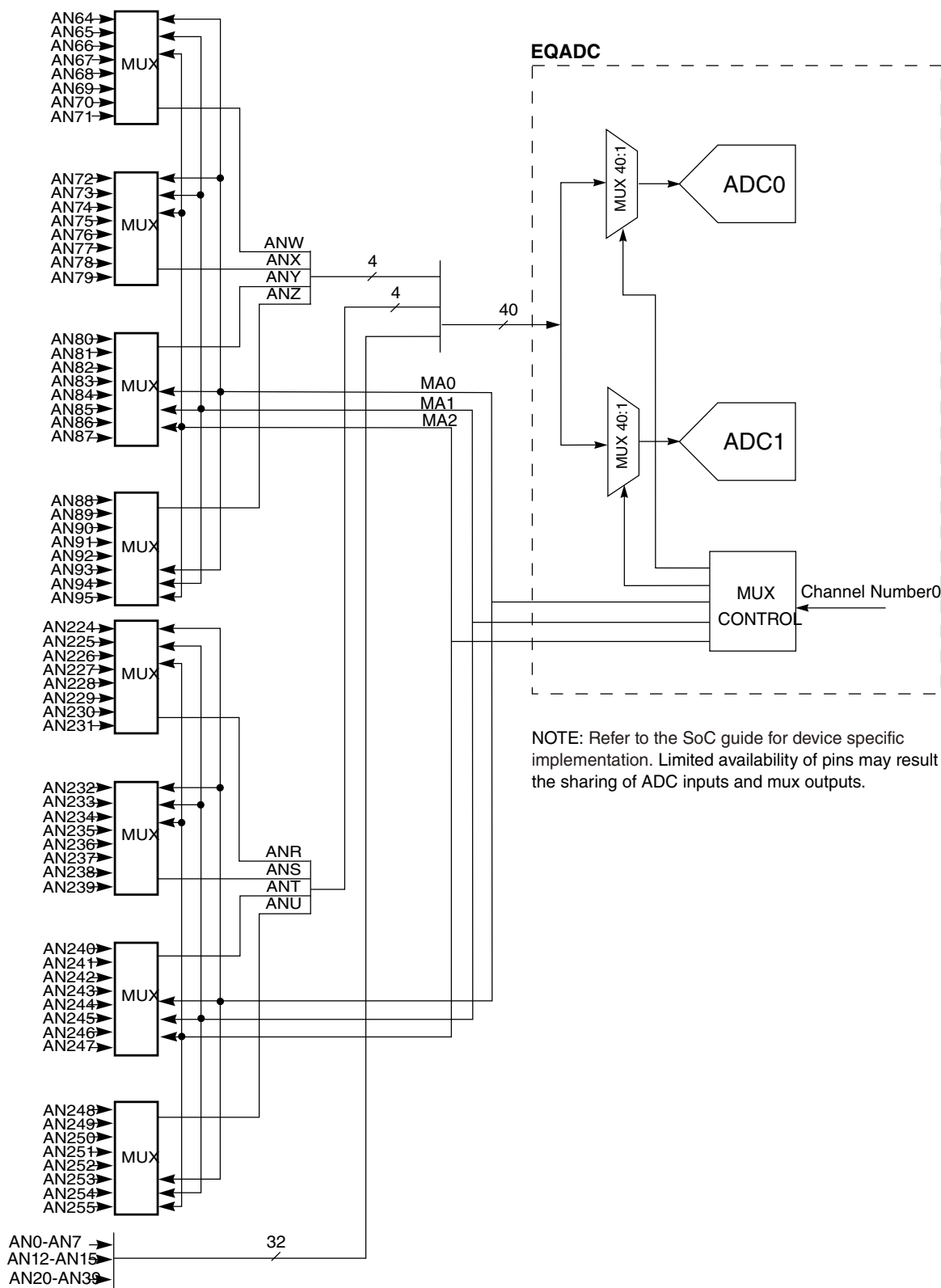


Figure 29-145. Example of external multiplexing



## 29.6.8 EQADC DMA/interrupt request

Table 29-141 lists methods to generate interrupt requests in the EQADC queuing control and triggering control. The DMA/interrupt request select bits and the DMA/interrupt enable bits are described in [EQADC Interrupt and DMA Control Register 0 \(EQADC\\_IDCR0\)](#) and the interrupt flag bits are described in [EQADC FIFO and Interrupt Status Register \(EQADC\\_FISR<sub>n</sub>\)](#). Figure 29-146 depicts all interrupts and DMA requests generated by the EQADC. For details, see [EQADC FIFO and Interrupt Status Register \(EQADC\\_FISR<sub>n</sub>\)](#) and [EQADC Interrupt and DMA Control Register 0 \(EQADC\\_IDCR0\)](#).

**Table 29-141. EQADC FIFO interrupt summary**

| Interrupt                                     | Condition                                                               | Clearing mechanism                                       |
|-----------------------------------------------|-------------------------------------------------------------------------|----------------------------------------------------------|
| Non Coherency Interrupt                       | NClEx = 1<br>NCFx = 1                                                   | Clear NCFx bit by writing a "1" to the bit.              |
| Result FIFO Overflow Interrupt <sup>1</sup>   | RFOlEx = 1<br>RFOF <sub>x</sub> = 1                                     | Clear RFOF <sub>x</sub> bit by writing a "1" to the bit. |
| Command FIFO Underflow Interrupt <sup>1</sup> | CFUIEx = 1<br>CFUF <sub>x</sub> = 1                                     | Clear CFUF <sub>x</sub> bit by writing a "1" to the bit. |
| Result FIFO Drain Interrupt                   | RFDE <sub>x</sub> = 1<br>RFDS <sub>x</sub> = 0<br>RFDF <sub>x</sub> = 1 | Clear RFDF <sub>x</sub> bit by writing a "1" to the bit. |
| Command FIFO Fill Interrupt                   | CFFEx = 1<br>CFFS <sub>x</sub> = 0<br>CFFF <sub>x</sub> = 1             | Clear CFFF <sub>x</sub> bit by writing a "1" to the bit. |
| End of Queue Interrupt                        | EOQlEx = 1<br>EOQF <sub>x</sub> = 1                                     | Clear EOQF <sub>x</sub> bit by writing a "1" to the bit. |
| Pause Interrupt                               | PIEx = 1<br>PF <sub>x</sub> = 1                                         | Clear PF <sub>x</sub> bit by writing a "1" to the bit.   |
| Trigger Overrun Interrupt <sup>1</sup>        | TORlEx = 1<br>TORF <sub>x</sub> = 1                                     | Clear TORF <sub>x</sub> bit by writing a "1" to the bit. |

1. Apart from generating an independent interrupt request for when a RFIFO Overflow Interrupt, a CFIFO Underflow Interrupt, and a CFIFO Trigger Overrun Interrupt occurs, the EQADC also provides a combined interrupt request at which these requests from ALL CFIFOs are ORed. Refer to [Figure 1](#) for details.

Table 29-142 describes a list of methods to generate DMA requests in the EQADC. For details, see [EQADC FIFO and Interrupt Status Register \(EQADC\\_FISR<sub>n</sub>\)](#) and [EQADC Interrupt and DMA Control Register 0 \(EQADC\\_IDCR0\)](#).

**Table 29-142. EQADC FIFO DMA Summary**

| DMA Request                                   | Condition                           | Clearing Mechanism                                                                                               |
|-----------------------------------------------|-------------------------------------|------------------------------------------------------------------------------------------------------------------|
| Result FIFO Drain DMA <sup>1</sup><br>Request | RFDEx = 1<br>RFDSx = 1<br>RFDFx = 1 | The EQADC automatically clears the RFDFx when RFIFOx becomes empty. Writing "1" to the RFDFx bit is not allowed. |
| Command FIFO Fill DMA<br>Request              | CFFEx = 1<br>CFFSx = 1<br>CFFFx = 1 | The EQADC automatically clears the CFFFx when CFIFOx becomes full. Writing "1" to the CFFFx bit is not allowed.  |

1. Additional readings from RFIFOx (not from DMA control) can clear the RFDFx bit and can lock the DMAx circuit to a locked state. The DMAx request remains ON, despite RFDFx is cleared.

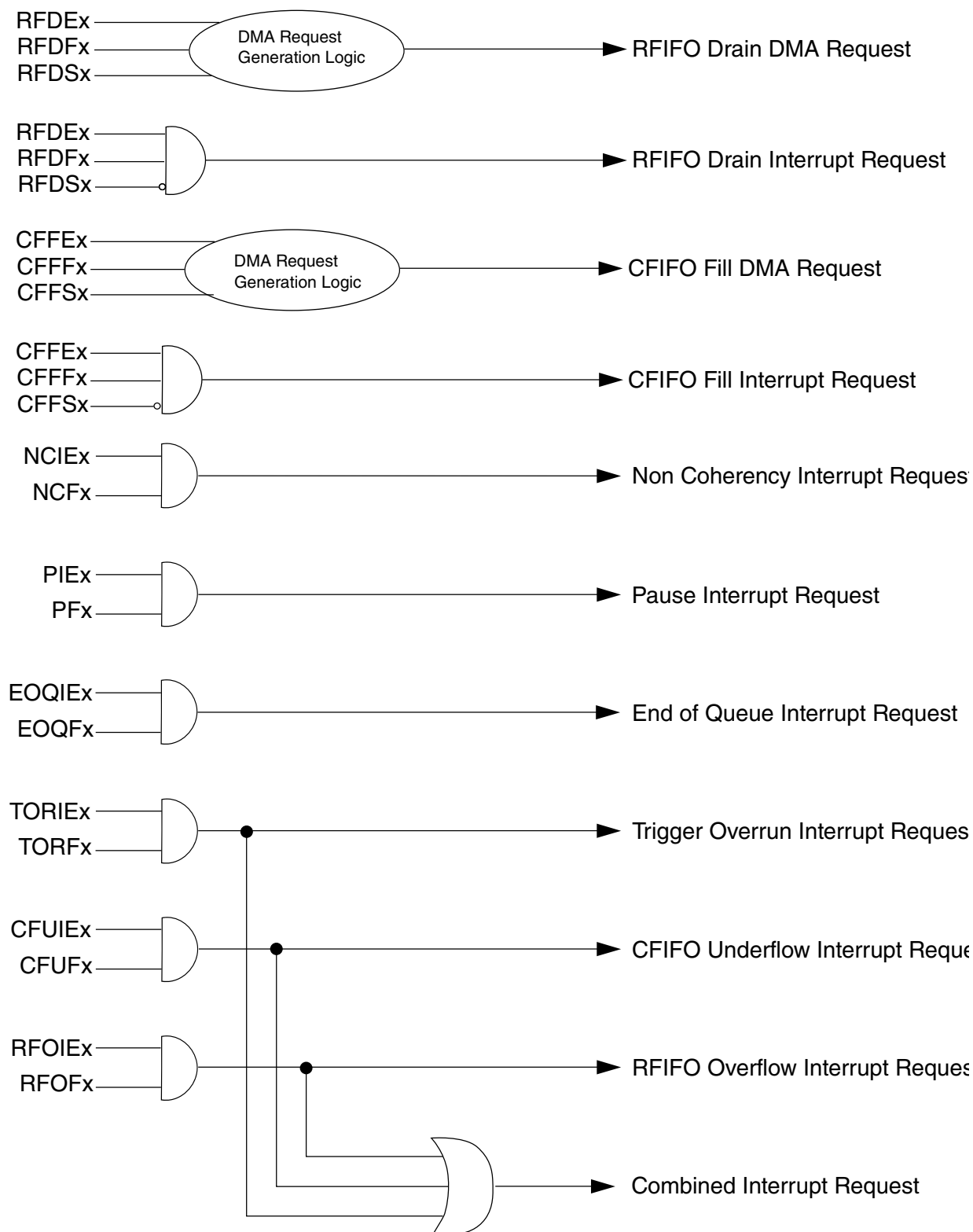


Figure 29-146. EQADC DMA and interrupt requests

## 29.6.9 EQADC Parallel Side Interface (PSI) sub-block

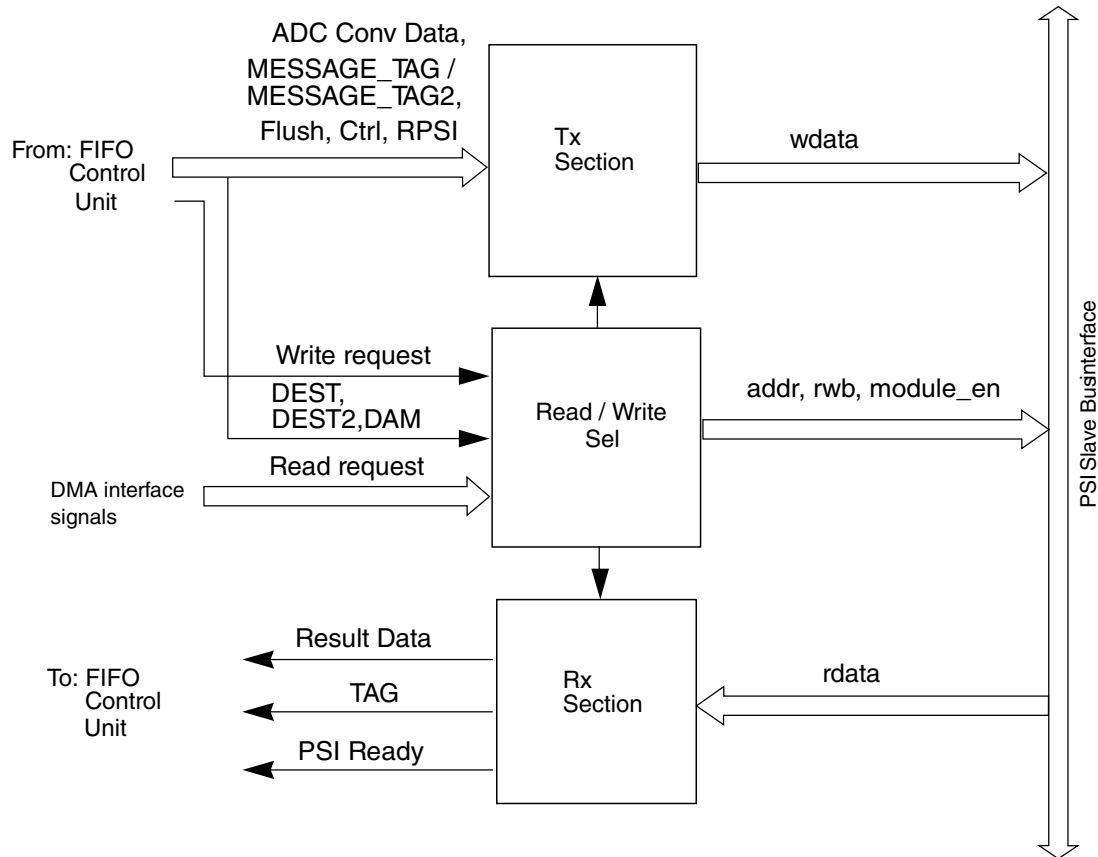


**Figure 29-147. EQADC Parallel Side Interface block diagram**

The EQADC PSI sub-block allows communication between the EQADC and the companion modules through a local slave bus that has the EQADC block as the master and the companion modules as the slaves. However, due to implementation constraints, a limited number of addresses is defined and a maximum of 15 different slave blocks to be enabled.

The conversion result goes through the Result Formatting and Calibration sub-block and the corresponding MESSAGE\_TAG / MESSAGE\_TAG2 and DEST/DEST2 fields are decoded to decide the destination of the conversion result data. When the DEST field is not zero as described in [Alternate Configuration 1-14 Control Registers \(ADC\\_ACR1-14\)](#), the MESSAGE\_TAG bits, control bits, and the conversion result data are sent to the transmission section of the PSI sub-block. This set of data is processed and sent to the corresponding on-chip companion module.

The configuration bit DAM is described in [EQADC Module Configuration Register \(EQADC\\_MCR\)](#). The DEST2/MESSAGE\_TAG2 fields are only used as defined in [Extended Alternate Configuration 1-14 Control Registers \(ADC\\_EACR1-14\)](#) when

DAM=1. DAM bit selects between two different "tap modes", each defining a different way to select the second destination. When DAM=0 the DEST field can set the transmission section to send the same result data / MESSAGE\_TAG to two destinations. See [Table 29-120](#) in [Alternate Configuration 1-14 Control Registers \(ADC\\_ACR1-14\)](#). When DAM=1, the two destinations are selected as described in [Table 29-124](#).

The companion module can also send back to EQADC the result of some processing in the received data. In this case, the receiver section of the PSI sub-block treats this request and receives the result data with the corresponding MESSAGE\_TAG or TAG field. This TAG is used by the decoder to select which RFIFO will be written with the received companion module data.

A companion module with two PSIs can be connected to two EQADCs. For this case, there is the bit field RPSI described in [Alternate Configuration 1-14 Control Registers \(ADC\\_ACR1-14\)](#) and the bit field RPSI2 described in [Alternate Configuration 1-14 Control Registers \(ADC\\_ACR1-14\)](#) that control the routing of the companion module output result to any PSI.

### 29.6.9.1 Input / output signals description

The information content of the input and the output data buses from/to the companion modules are described in [Figure 29-148](#). The input data format is used in the Read cycle and corresponds to the data supplied by some companion module. The output data format is used in the Write cycle of this interface block and contains information / controls to some companion module.

## On-chip ADC configuration and control

Read (Input) / Write (Output) Data Buses Content

|        |   |   |   |   |   |   |   |   |       |           |    |      |                  |    |    |    |
|--------|---|---|---|---|---|---|---|---|-------|-----------|----|------|------------------|----|----|----|
|        | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8     | 9         | 10 | 11   | 12               | 13 | 14 | 15 |
| R      |   |   |   |   |   |   |   |   | 0     | 0         | 0  | 0    | TAG[0:3]         |    |    |    |
| W      |   |   |   |   |   |   |   |   | FLUSH | CTRL[0:1] |    | RPSI | MESSAGE_TAG[0:3] |    |    |    |
| RESET: | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0     | 0         | 0  | 0    | 0                | 0  | 0  | 0  |

|        |                       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|--------|-----------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
|        | 16                    | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R      | RESULT_DATA[0:15]     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W      | ADC_CONV_RESULT[0:15] |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| RESET: | 0                     | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

 = Unimplemented or Reserved

**Figure 29-148. PSI input and output data buses content**

**Table 29-143. PSI input and output data buses content — field descriptions**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| FLUSH | Master block Flush request/control bit<br><br>The FLUSH signal is used to request a Flush flow in the slave block. More details is presented in <a href="#">Conversion command format for alternate configurations</a> .<br>0 No flush request.<br>1 Flush request                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| RPSI  | Return Parallel Side Interface<br><br>The RPSI bit is used in the companion module that has two PSI interfaces (PSI0 and PSI1) which can be connected to two eQADCs. It specifies in the companion module if the data received from one PSI should be transmitted to the same PSI connected to this eQADC or to the other one. This bit is defined by the bit field RPSI described in <a href="#">Alternate Configuration 1-14 Control Registers (ADC_ACR1-14)</a> or by the bit field RPSI2 described in <a href="#">Alternate Configuration 1-14 Control Registers (ADC_ACR1-14)</a> .<br>0 Results from this sample must be transmitted back using the same PSI (eQADC).<br>1 Results from this sample must be transmitted to the other PSI (eQADC) |
| CTRL  | Control bits<br><br>This field is used for the control of the companion module.<br>00 INH_RET or PREFILL - This mode indicates that the companion module should not send back some RESULT_DATA corresponding to the accompanying ADC_CONV_RESULT data. When the slave module is the Decimation Filter, this control enables the PREFILL filter mode.                                                                                                                                                                                                                                                                                                                                                                                                   |

*Table continues on the next page...*

**Table 29-143. PSI input and output data buses content — field descriptions (continued)**

| Field                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                       | <p>01 CONVERSION RESULT - It indicates that the ADC_CONV_RESULT field should be treated as a valid sample data. This control mode is useful to Decimation Filter to put the filter em normal mode instead of prefill mode.</p> <p>10 TIME STAMP o REGISTER READ - A time stamp indicates that the ADC_CONV_RESULT field should follow a bypass flow in the companion module, returning back to the EQADC without any modification.</p> <p>11 Reserved</p>                                                                                                                                                                                            |
| MESSAGE_TAG[0:3]      | <p>Message tag bits field</p> <p>This field indicates the RFIFO destination associated with the ADC_CONV_RESULT sample. This value is stored by the companion module and is used to address the destination RFIFO register when a RESULT_DATA is generated due to that ADC_CONV_RESULT sample.</p> <p><b>Note:</b> In case the companion module only receives data from eQADC and does not return any result to eQADC RFIFOs, the MESSAGE_TAG field can be used as a destination router in the companion module. The eQADC module can send conversion results to sub-blocks in the companion module that are addressed by the MESSAGE_TAG value.</p> |
| TAG[0:3]              | <p>Companion module tag bits</p> <p>This bit field is used to address the appropriate destination RFIFO in the EQADC for the accompanying RESULT_DATA bits. However, it can be used too in the companion module to select the destination of the conversion result data.</p> <p>In eQADC application, this is used to address the appropriate RFIFO in the eQADC block. In this case, the possible values are only from 0000 to 0101.</p>                                                                                                                                                                                                            |
| ADC_CONV_RESULT[0:15] | <p>ADC Conversion Result Data</p> <p>This bit field is the ADC conversion result data after passing through the calibration and formatting block.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| RESULT_DATA[0:15]     | <p>Companion Module Result Data</p> <p>This bit field corresponds to the companion module data processing result to EQADC.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |

### 29.6.9.2 PSI Transmitter / Write Section

The transmission sub-block formats the data bus from RFIFO control sub-block to send to the PSI slave wdata bus. The transmission data is registered and its content is described in [Input / output signals description](#). The transmission has higher priority than reception. This is done to avoid the use of memory to store transmission data and it is not used waiting time for transmission.

The destination companion module for the transmission data is obtained by decoding the DEST[0:3] and DEST2 bits. The not null decimal value of DEST[0:3] or DEST2 is used to set one or two corresponding module enable signals as already described. For example, DEST[0:3] value equal to 0x1 that corresponds to the decimal value 1 is going to set only the module enable 1. All other module enable from 2 to 15 are not set due to DEST.

However, using the DEST2 value there are cases that two module enables are set, as described in [Table 29-120](#) in [Alternate Configuration 1-14 Control Registers \(ADC\\_ACR1-14\)](#).

### **29.6.9.3 PSI Receiver / Read Section**

The receiver sub-block receives data from some companion module using the PSI slave bus interface. The companion module sends a read request to EQADC using the DMA read request line. The PSI logic sends a read command if there is no transmission request. The received data has the structure described in [Input / output signals description](#).

The receiver section can receive read request from several companion modules at the same time. A prioritization scheme is used to decide which request should be served first. The request with lower index has higher priority.

#### **Note**

In case of receiving data (D1, D2) and timestamp (T1, T2) from two companion modules with the same TAG at almost the same time, the RFIFOx can receive the pair data/timestamp out of order due to the present prioritization logic in the PSI receiver. An expected sequence (D1 T1 D2 T2) can be changed to (D1 D2 T2 T1) if D2/T2 has higher priority than T2. [Table 29-124](#) presents some cases that presents this inversion.

## **29.6.10 Analog sub-block**

### **29.6.10.1 Analog to digital converter (ADC)**

#### **29.6.10.1.1 ADC architecture**



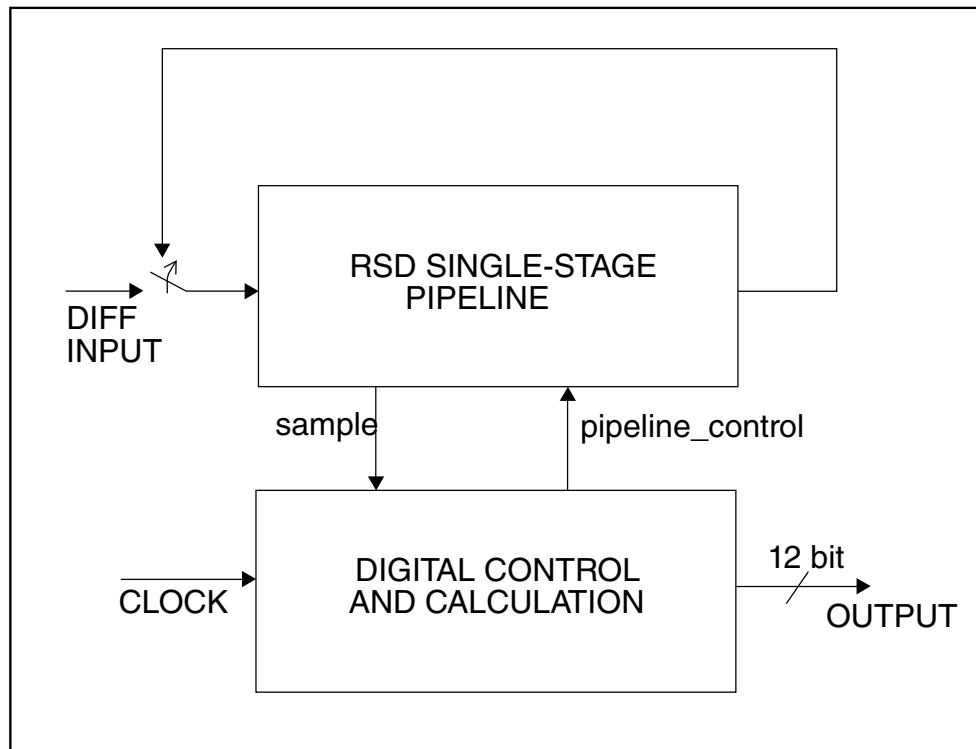


**Figure 29-149. RSD ADC block diagram**

The RSD Cyclic ADC consists of two main portions, the analog RSD Stage, and the digital control and calculation block, as shown in [Figure 29-149](#). To begin an analog to digital conversion, a differential input is passed into the analog RSD stage. The signal is passed through the RSD stage, and then from the RSD stage output, back to its input to be passed again. To complete a 12-bit conversion, the signal must pass through the RSD stage 12 times. For 10-bit and 8-bit resolution, the signal must pass 10 or 8 times through the RSD. Each time an input signal is read into the RSD stage, a digital sample is taken by the digital control/calculation block. The digital control/calculation block uses this sample to tell the analog block how to condition the signal. The digital block also saves each successive sample and adds them according to the RSD algorithm at the end of the entire conversion cycle.

#### 29.6.10.1.2 RSD Overview

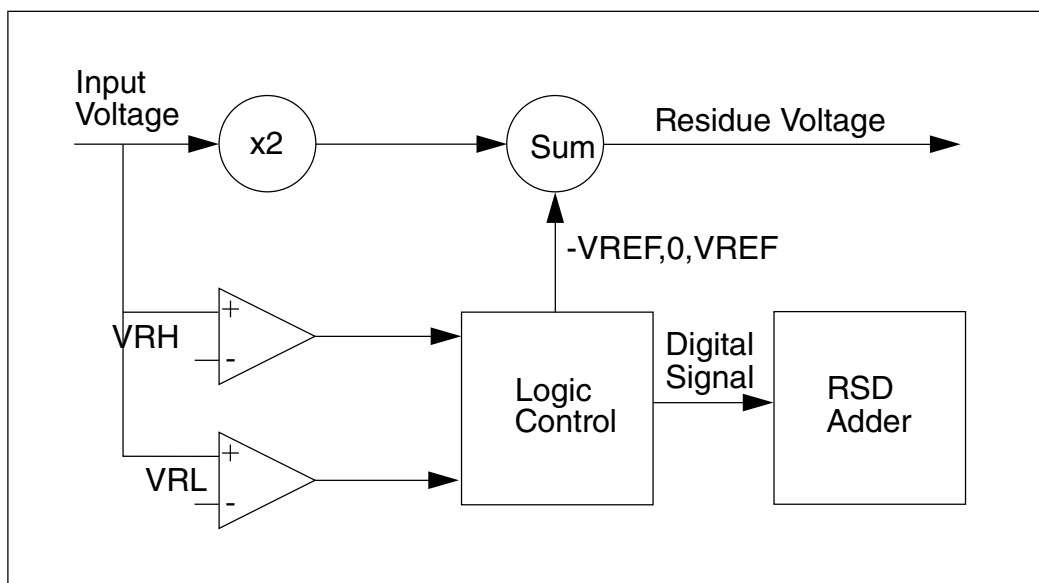


**Figure 29-150. RSD stage block diagram**

On each pass through the RSD stage, the input signal will be multiplied by exactly two, and summed with either  $-V_{REF}$ , 0, or  $V_{REF}$ , depending on the Logic Control. The Logic Control will determine  $-V_{REF}$ , 0, or  $V_{REF}$  depending on the two comparator inputs. As the Logic Control sets the summing operation, it also sends a digital value to the RSD adder. Each time an analog signal passes through the RSD single-stage, a digital value is collected by the RSD adder. At the end of an entire AD conversion cycle, the RSD adder uses these collected values to calculate the 12-bit/10-bit/8-bit digital output.

[Figure 29-151](#) shows the transfer function for the RSD stage. Note how the digital value (a, b) is dependent on the two comparator inputs.

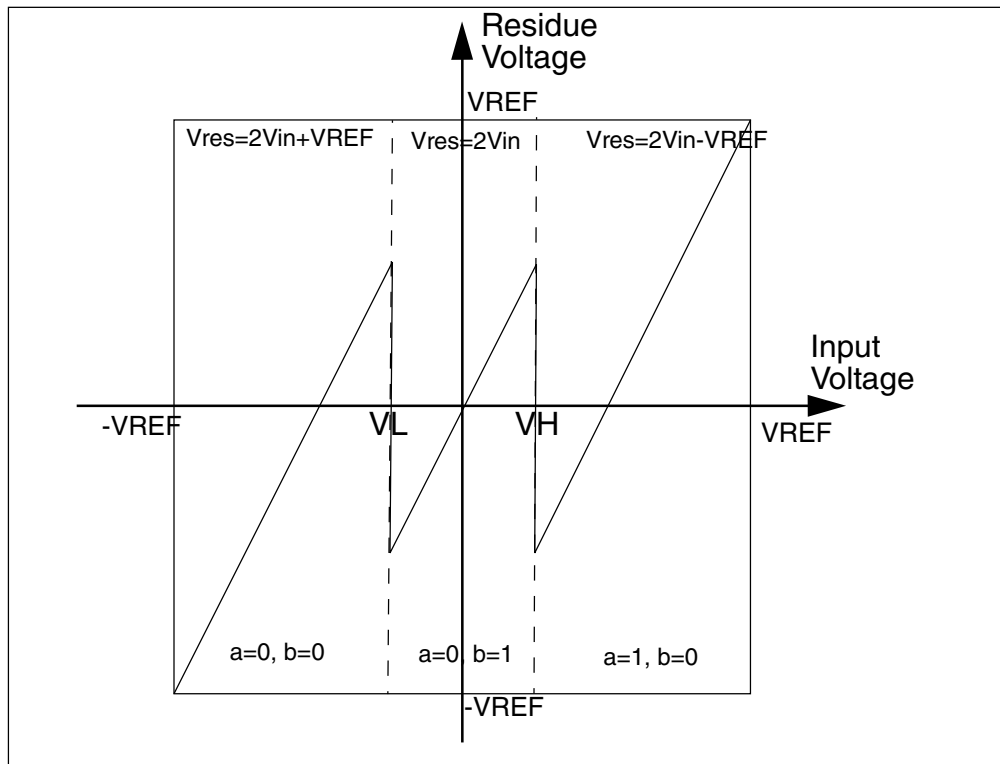


**Figure 29-151. RSD Stage Transfer Function**

In each pass through the RSD stage, the residue will be sent back to be the new input, and the digital signals, *a* and *b*, will be stored. For the 12-bit ADC, input signal is sampled during the input phase, and after each of the 12 passes through the RSD stage. Thus, 13 total *a* and *b* values are collected. Upon collecting all these values, they will be added according to the RSD algorithm to create the 12-bit digital representation of the original analog input. The bits are added in the following manner:

#### 29.6.10.1.3 RSD adder

The array, *s1* to *s12*, will be the digital output of the RSD ADC with *s1* being the MSB and *s12* being the LSB.

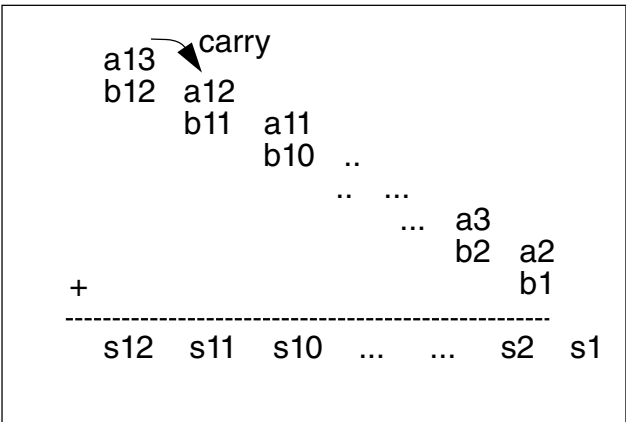


Figure 29-152. RSD adder

#### 29.6.10.1.4 Variable Gain Amplification (VGA) for pre-gain

The VGA starts after sampling completes. It is enabled by a 2-bit signal `PRE_GAIN` described in [Alternate Configuration 1-14 Control Registers \(ADC\\_ACR1-14\)](#).

The ADC takes 2, 8, 64 or 128 clock cycles to do sampling which is selected by the `LST[0:1]` field in the conversion command message. After the sampling, if 2x VGA is enabled, there is a 2x gain stage without comparison before the regular conversion cycles. When 4x VGA is enabled, there are the 2x gain stage without comparison by 2 times before the normal conversion processing.

## 29.7 Initialization/application information

### 29.7.1 Multiple queues control setup example

This section provides an example of how to configure multiple CQueues. [Table 29-144](#) describes how each CQueue can be used for a different application. Also documented in this section are general guidelines on how to initialize the on-chip ADCs and how to configure the CQueues and the EQADC.

Table 29-144. Application of each CQueue

| CQueue number | CQueue type                       | Running speed                                                          | Number of contiguous conversions | Example                                     |
|---------------|-----------------------------------|------------------------------------------------------------------------|----------------------------------|---------------------------------------------|
| 0             | Very fast burst time-based CQueue | every 2 $\mu$ s for 200 $\mu$ s; pause for 300 $\mu$ s and then repeat | 2                                | Injector current profiling                  |
| 1             | Fast hardware-triggered CQueue    | every 900 $\mu$ s                                                      | 3                                | Current sensing of PWM controlled actuators |

Table continues on the next page...

**Table 29-144. Application of each CQueue (continued)**

| CQueue number | CQueue type                       | Running speed     | Number of contiguous conversions | Example                                   |
|---------------|-----------------------------------|-------------------|----------------------------------|-------------------------------------------|
| 2             | Fast repetitive time-based CQueue | every 2 ms        | 8                                | Throttle position                         |
| 3             | Software-triggered CQueue         | every 3.9 ms      | 3                                | Command triggered by software strategy    |
| 4             | Repetitive angle-based CQueue     | every 625 $\mu$ s | 7                                | Airflow read every 30 degrees at 8000 RPM |
| 5             | Slow repetitive time-based CQueue | every 100 ms      | 10                               | Temperature sensors                       |

### 29.7.1.1 EQADC initialization

The following steps provide an example about how to configure the EQADC controls and how to initialize the on-chip ADCs. In this example, all conversion commands will be transferred through CFIFO0.

1. Load all required configuration commands in the RAM in such way that they form a queue; this data structure will be referred below as CQueue0. [Figure 29-153](#) shows an example of a CQueue able to configure the on-chip ADCs. Although, this example uses the DMAC to store commands in CFIFO0, configuration commands could have also been directly written to the CFIFO0 push register.
2. Select source driving EQADC hardware trigger ports (ETRIG). Before proceeding to next step, allow some time (minimum of two system clocks - filter period is set to minimum after reset) so that the logic level at the source is filtered and reaches the EQADC control logic.

#### Note

ETRIG ports could be driven by an external pin or by the output port of other blocks in the device, such as timers. In order to avoid unexpected triggering of CFIFOs in hardware trigger modes, the source driving the ETRIG port must be selected and set to a known logic level before putting the CFIFOs into the WAITING FOR TRIGGER state.

The trigger filter bypass control inputs must be set considering the characteristics of the trigger signal. A particular case to assert the bypass control is when a device's internal signal with one clock width pulse is used.

- 3. Configure [EQADC External Trigger Digital Filter Register \(EQADC\\_ETDFR\)](#).
- 4. Configure the DMAC to transfer data from CQueue0 to CFIFO0 in the EQADC.
- 5. Configure [EQADC Interrupt and DMA Control Register 0 \(EQADC\\_IDCR0\)](#).
  - Set CFFS0 to configure the EQADC to generate a DMA request to load commands from CQueue 0 to the CFIFO0.
  - Set CFFE0 to enable the EQADC to generate a DMA request to transfer commands from CQueue0 to CFIFO0; Command transfers from the RAM to the CFIFO0 will start immediately.
  - Set EOQIE0 to enable the EQADC to generate an interrupt after transferring all of the commands of CQueue0 through CFIFO0.
- 6. Configure [EQADC CFIFO Control Register 0 \(EQADC\\_CFCR0\)](#).
  - Write 0b0001 to the MODE0 field in EQADC\_CFCR0 to program CFIFO0 for software single-scan mode.
  - Write "1" to SSE0 to assert SSS0 and trigger CFIFO0.
- 7. Since CFIFO0 is in single-scan software mode and it is also the highest priority CFIFO, the EQADC starts to transfer configuration commands to the on-chip ADCs.
- 8. When all of the configuration commands have been transferred, CF0 in [EQADC FIFO and Interrupt Status Register \(EQADC\\_FISRn\)](#) will be set. The EQADC generates a End of Queue interrupt. The initialization procedure is complete.

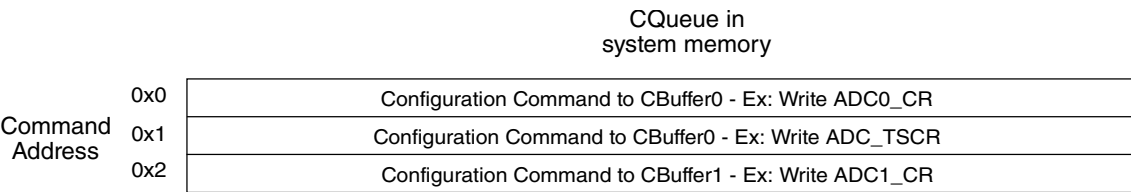


Figure 29-153. Example of a CQueue configuring the on-chip ADCs

The initialization procedure described above does not generate ADC clocks that are in phase because the timing at which the ADC0/1\_EN bits, in [ADC0/1 Control Registers \(ADC0\\_CR and ADC1\\_CR\)](#), are set is different. Below follows an example on how to simultaneously set these bits so that in-phase ADC clocks are generated. In this example, ADC0/1\_CLK are configured to the same frequency.

1. Push an ADC0\_CR write configuration command in CFIFO0 that enables ADC0 (ADC0\_EN=1) and that sets the ADC0\_CLK\_PS to an appropriate value. For example, 0x80800801.
2. Push an ADC1\_CR write configuration command in CFIFO1 that enables ADC1 (ADC1\_EN=1) and that sets the ADC1\_CLK\_PS to an appropriate value. For example, 0x82800801.
3. Configure CFIFO0 and CFIFO1 to single scan software trigger mode and simultaneously trigger them by writing 0x04100410 to the EQADC\_CFCR0 register - see [EQADC CFIFO Control Register 0 \(EQADC\\_CFCR0\)](#).

### 29.7.1.2 Configuring EQADC for applications

This section provides an example based on the applications in [Table 29-144](#). The example describes how to configure multiple CQueues to be used for those applications and provides a step-by-step procedure to configure the EQADC and the associated CQueue structures. In the example, the "Fast hardware-triggered CQueue", described on the second row of [Table 29-144](#), will have its commands transferred to CBuffer1; the conversion commands will be executed by ADC1. The generated results will be returned to RFIFO3 before being transferred to the RQueues in the RAM by the DMAC.

#### Note

There is no fixed relationship between CFIFOs and RFIFOs with the same number. The results of commands being transferred through CFIFO1 can be returned to any RFIFO, regardless of its number. The destination of a result is determined by the MESSAGE\_TAG field of the command that requested the result. See [Message format in EQADC](#) for details.

Step One: Setup the CQueues and RQueues.

1. Load the RAM with configuration and conversion commands. [Figure 29-154](#) is an example of how CQueue1 commands should be set.

- Each trigger event will cause four commands to be executed. When the EQADC detects the Pause bit asserted, it will wait for another trigger to restart transferring commands from the CFIFO.
- At the end of the CQueue, the "EOQ" bit is asserted as shown in [Figure 29-154](#).
- Results will be returned to RFIFO3 as specified in the MESSAGE\_TAG field of commands.

## 2. Reserve memory space for storing results.

| Bit Name     | 0  | 1     | 2   | 3        | 4  | 5  | 6        | 7                   | 8                                                        | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|--------------|----|-------|-----|----------|----|----|----------|---------------------|----------------------------------------------------------|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
|              | EQ | PAUSE | REP | RESERVED | FB | BN | RESERVED | MESSAGE TAG         | ADC COMMAND                                              |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| CMD 1        | 0  | 0     | 0   | 0        | 0  | 1  | 0        | 0b0011              | Conversion Command                                       |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| CMD 2        | 0  | 0     | 0   | 0        | 0  | 1  | 0        | 0b0011              | Conversion Command                                       |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| CMD 3        | 0  | 0     | 0   | 0        | 0  | 1  | 0        | 0b0011              | Conversion Command                                       |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| CMD 4        | 0  | 1     | 0   | 0        | 0  | 1  | 0        | 0b0011 <sup>1</sup> | Configure peripheral device for next conversion sequence |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| CMD 5        | 0  | 0     | 0   | 0        | 0  | 1  | 0        | 0b0011              | Conversion Command                                       |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| CMD 6        | 0  | 0     | 0   | 0        | 0  | 1  | 0        | 0b0011              | Conversion Command                                       |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| CMD 7        | 0  | 0     | 0   | 0        | 0  | 1  | 0        | 0b0011              | Conversion Command                                       |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| CMD 8        | 0  | 1     | 0   | 0        | 0  | 1  | 0        | 0b0011 <sup>1</sup> | Configure peripheral device for next conversion sequence |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|              |    |       |     | 0        |    |    |          |                     | etc. ....                                                |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| CMD EQ       | 1  | 0     | 0   | 0        | 0  | 1  | 0        | 0b0011              | EOQ Message                                              |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| CFIFO Header |    |       |     |          |    |    |          |                     | ADC Command                                              |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

Fields LST, TSR, FMT, and CHANNEL\_NUMBER are not showed for clarity.

<sup>1</sup> MESSAGE\_TAG field is only defined for read configuration commands.

**Figure 29-154. Example of CQueue commands**

Step Two: Configure the DMAC to handle data transfers between the CQueues/RQueues in RAM and the CFIFOs/RFIFOs in the EQADC.



1. For transferring, set the source address of the DMAC to point to the start address of CQueue1. Set the destination address of the DMAC to point to EQADC\_CFPR1. See [EQADC CFIFO Push Register \(EQADC\\_CFPRn\)](#).
2. For receiving, set the source address of the DMAC to point to EQADC\_RFPR3. See [EQADC Result FIFO Pop Register \(EQADC\\_RFPRn\)](#). Set the destination address of the DMAC to point to the starting address of RQueue1.

Step Three: Configure the EQADC Control Registers.

3. Configure [EQADC Interrupt and DMA Control Register 0 \(EQADC\\_IDCR0\)](#).
  - Set EOQIE1 to enable the End of Queue Interrupt request.
  - Set CFFS1 and RFDS3 to configure the EQADC to generate DMA requests to push commands into CFIFO1 and to pop result data from RFIFO3.
  - Set CFINV1 to invalidate the contents of CFIFO1.
  - Set RFDE3 and CFFE1 to enable the EQADC to generate DMA requests. Command transfers from the RAM to the CFIFO1 will start immediately.
  - Set RFOIE3 to indicate if RFIFO3 overflows.
  - Set CFUIE1 to indicate if CFIFO1 underflows.
4. Configure MODE1 to continuous-scan rising edge external trigger mode in [EQADC CFIFO Control Register 0 \(EQADC\\_CFCR0\)](#).

Step Four: Command transfer to ADCs and Result data reception.

When an external rising edge event occurs for CFIFO1, the EQADC automatically will begin transferring commands from CFIFO1 when it becomes the highest priority CFIFO trying to send commands to CBuffer1. The received results will be placed in RFIFO3 and then moved to RQueue1 by the DMAC.

## 29.7.2 EQADC/DMAC interface

This section provides an overview about the EQADC/DMAC interface and general guidelines about how the DMAC should be configured in order for it to correctly transfer data between the queues in system memory and the EQADC FIFOs.

### Note

Advanced DMACs provide more functionality than the ones discussed in this section.

### 29.7.2.1 CQueue/CFIFO transfers

In transfers involving CQueues and CFIFOs, the DMAC moves data from a queued source to a single destination as showed in [Figure 29-155](#). The location of the data to be moved is indicated by the source address, and the final destination for that data, by the destination address. The DMAC contains a data structure containing these addresses and other parameters used in the control of data transfers. For every DMA request issued by the EQADC, the DMAC has to be configured to transfer a single command (32-bit data) from the CQueue, pointed to by the source address, to the CFIFO push register, pointed to by the destination address. After the service of a DMA request is completed, the source address has to be updated to point to the next valid command. The destination address remains unchanged. When the last command of a queue is transferred one of the following actions is recommended. Refer to the DMAC block guide for details about how this functionality is supported.

- The corresponding DMA channel should be disabled. This might be desirable for CFIFOs in single scan mode.
- The source address should be updated to pointed to a valid command which can be the first command in the queue that has just been transferred (cyclic queue), or the first command of any other CQueue. This is desirable for CFIFOs in continuous scan mode, and at some cases, for CFIFOs in single scan mode.

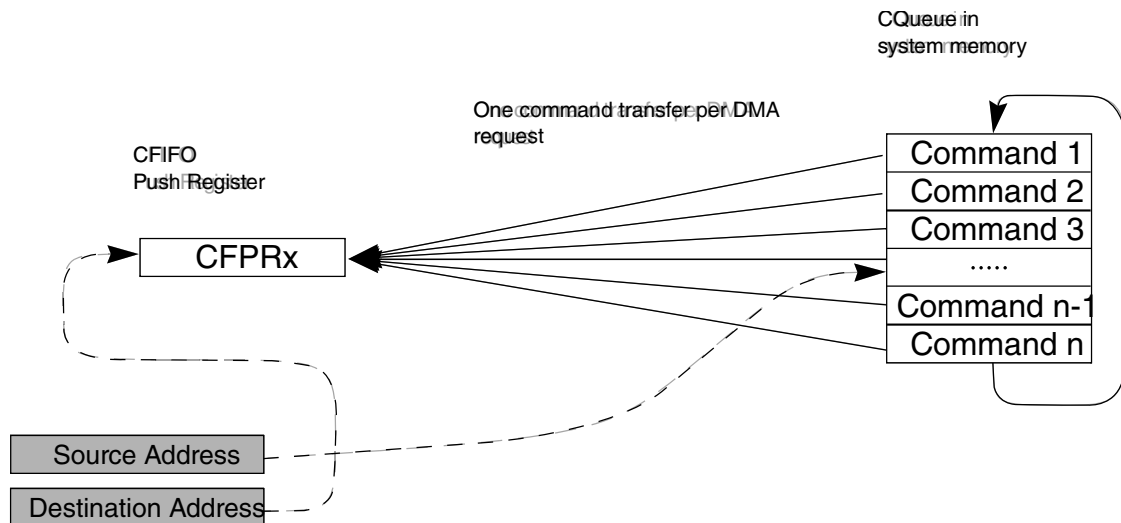


Figure 29-155. CQueue/CFIFO interface

### 29.7.2.2 RQueue/RFIFO transfers

In transfers involving RQueues and RFIFOs, the DMAC moves data from a single source to a queue destination as showed in Figure 29-156. The location of the data to be moved is indicated by the source address, and the final destination for that data, by the destination address. For every DMA request issued by the EQADC, the DMAC has to be configured to transfer a single result (16-bit data), pointed to by the source address, from the RFIFO pop register to the RQueue, pointed to by the destination address. After the service of a DMA request is completed, the destination address has to be updated to point to the location where the next 16-bit result will be stored. The source address remains unchanged. When the last expected result is written to the RQueue, one of the following actions is recommended. Refer to the DMAC block guide for details about how this functionality is supported.

- The corresponding DMA channel should be disabled.
- The destination address should be updated pointed to the next location where new coming results are stored, which can be the first entry of the current RQueue (cyclic queue), or the beginning of a new RQueue.

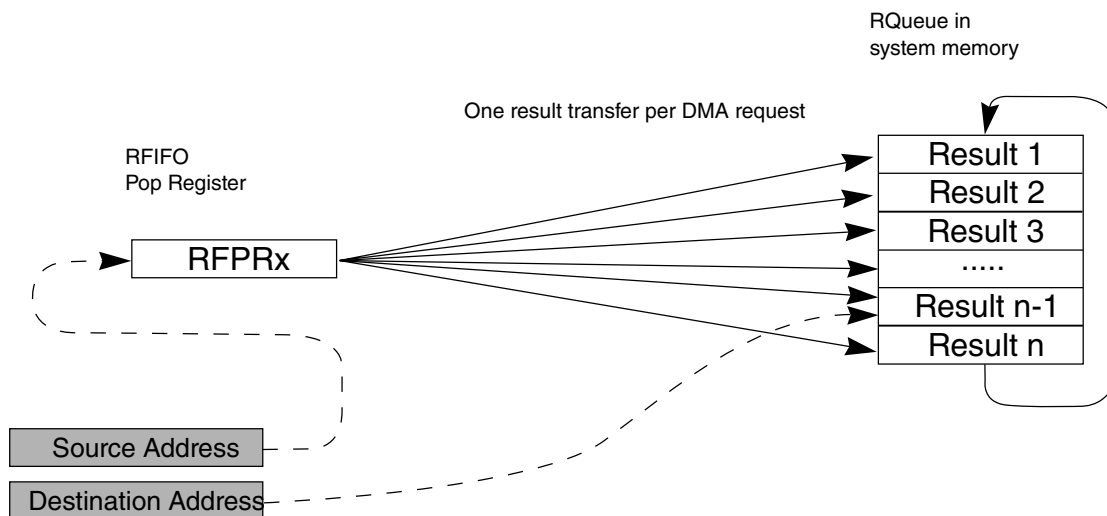


Figure 29-156. RQueue/RFIFO Interface

### 29.7.3 Sending immediate command setup example

In the EQADC, there is no immediate command register for sending a command immediately after writing to that register. However, a CFIFO can be configured to perform the same function as an immediate command register. The following steps illustrate how to configure CFIFO5 as an immediate command CFIFO. The results will be returned to RFIFO5.

1. Configure [EQADC Interrupt and DMA Control Register 0 \(EQADC\\_IDCR0\)](#).
  - Clear CFIFO Fill Enable5 (CFFE5 = 0) in EQADC\_IDCR2.
  - Clear CFIFO Underflow Interrupt Enable5 (CFUIE5 = 0) in EQADC\_IDCR2.
  - Clear RFDS5 to configure the EQADC to generate interrupt requests to pop result data from RFIFO5.
  - Set RFIFO Drain Enable5 (RFDE5 = 1) in EQADC\_IDCR2.
2. Configure [EQADC CFIFO Control Register 0 \(EQADC\\_CFCR0\)](#).
  - Write "1" to CFINV5 in EQADC\_FCR2. This will invalidate the contents of CFIFO5.
  - Set MODE5 to Continuous-Scan Software Trigger mode in EQADC\_CFCR2.
3. To transfer a command, write it to EQADC CFIFO Push Register 5 (EQADC\_CFPR5) with Message Tag = 0b0101. See [EQADC CFIFO Push Register \(EQADC\\_CFPRn\)](#).
4. Up to four commands can be queued in CFIFO5. Check the CFCTR5 status in EQADC\_FISR5 before pushing another command to avoid overflowing the CFIFO. See [EQADC FIFO and Interrupt Status Register \(EQADC\\_FISRn\)](#).
5. When the EQADC receives a conversion result for RFIFO5, it generates an interrupt request. RFIFO Pop Register 5 (EQADC\_RFPR5) can be popped to read the result. See [EQADC Result FIFO Pop Register \(EQADC\\_RFPRn\)](#).

## 29.7.4 Modifying queues

More CQueues may be needed than the six supported by the EQADC. These additional CQueues can be supported by interrupting command transfers from a configured CFIFO, even if it is TRIGGERED and transferring, modifying the corresponding CQueue in the RAM or associating another CQueue to it, and restarting the CFIFO. More details on disabling a CFIFO are described in [Disabled Mode](#).

1. Determine the resumption conditions when later resuming the scan of the CQueue at the point before it was modified.
  - Change MODEx in [EQADC CFIFO Control Register 0 \(EQADC\\_CFCR0\)](#) to Disabled. See [Disabled Mode](#) for a description of what happens when MODEx is changed to Disabled.

- Poll CFSx until it becomes IDLE in [EQADC CFIFO Status Register \(EQADC\\_CFSR\)](#).
  - Read and save TC\_CFx in [EQADC CFIFO Transfer Counter Register 0 \(EQADC\\_CFTCR0\)](#) for later resuming the scan of the queue. The TC\_CFx provides the point of resumption.
  - Since all result data may not have been stored in the appropriate RFIFO at the time MODEx is changed to disable, wait for all expected results to be stored in the RFIFO/RQueue before reconfiguring the DMAC to work with the modified RQueue. The number of results that must return can be estimated from the TC\_CFx value obtained above.
2. Disable the DMAC from responding to the DMA request generated by CFFFx and RFDFx in [EQADC FIFO and Interrupt Status Register \(EQADC\\_FISRn\)](#).
  3. Write "0x0000" to the TC\_CFx field.
  4. Load the new configuration and conversion commands into RAM. Configure the DMAC to support the new CQueue/RQueue, but do not configure it yet to respond to DMA requests from CFIFOx/RFIFOx.
  5. If necessary, modify [EQADC Interrupt and DMA Control Register 0 \(EQADC\\_IDCR0\)](#) to suit the modified CQueue.
  6. Write "1" to CFINVx in [EQADC CFIFO Control Register 0 \(EQADC\\_CFCR0\)](#) to invalidate the entries of CFIFOx. Perform any other modifications to EQADC\_CFCR except changing MODEx from Disabled.
  7. Configure the DMAC to respond to DMA requests generated by CFFFx and RFDFx.
  8. Change MODEx to the modified CFIFO operation mode. Write "1" to SSEx to trigger CFIFOx if MODEx is software trigger.

### 29.7.5 CQueue and RQueues usage

[Figure 29-157](#) is an example of CQueue and RQueue usage. It shows the CQueue0 commands requesting results that will be stored in RQueue0 and RQueue1, and CQueue1 commands requesting results that will be stored only in RQueue1. Some Command Messages request data to be returned from the on-chip ADC, but some only configure them and do not request returning data. When a CQueue contains both write and read commands like CQueue0, the CQueue and RQueue entries will not be aligned; as shown in [Figure 29-157](#), the result for the second command of CQueue0 is the first entry of RQueue0. The figure also shows that CQueue and RQueue entries can also become

unaligned even if all commands in a CQueue request data as CQueue1. CQueue1 entries became unaligned to RQueue1 entries because a result requested by the forth CQueue0 command was sent to RQueue1. This happens because the system can be configured so that several CQueues can have its results sent to a single RQueue.

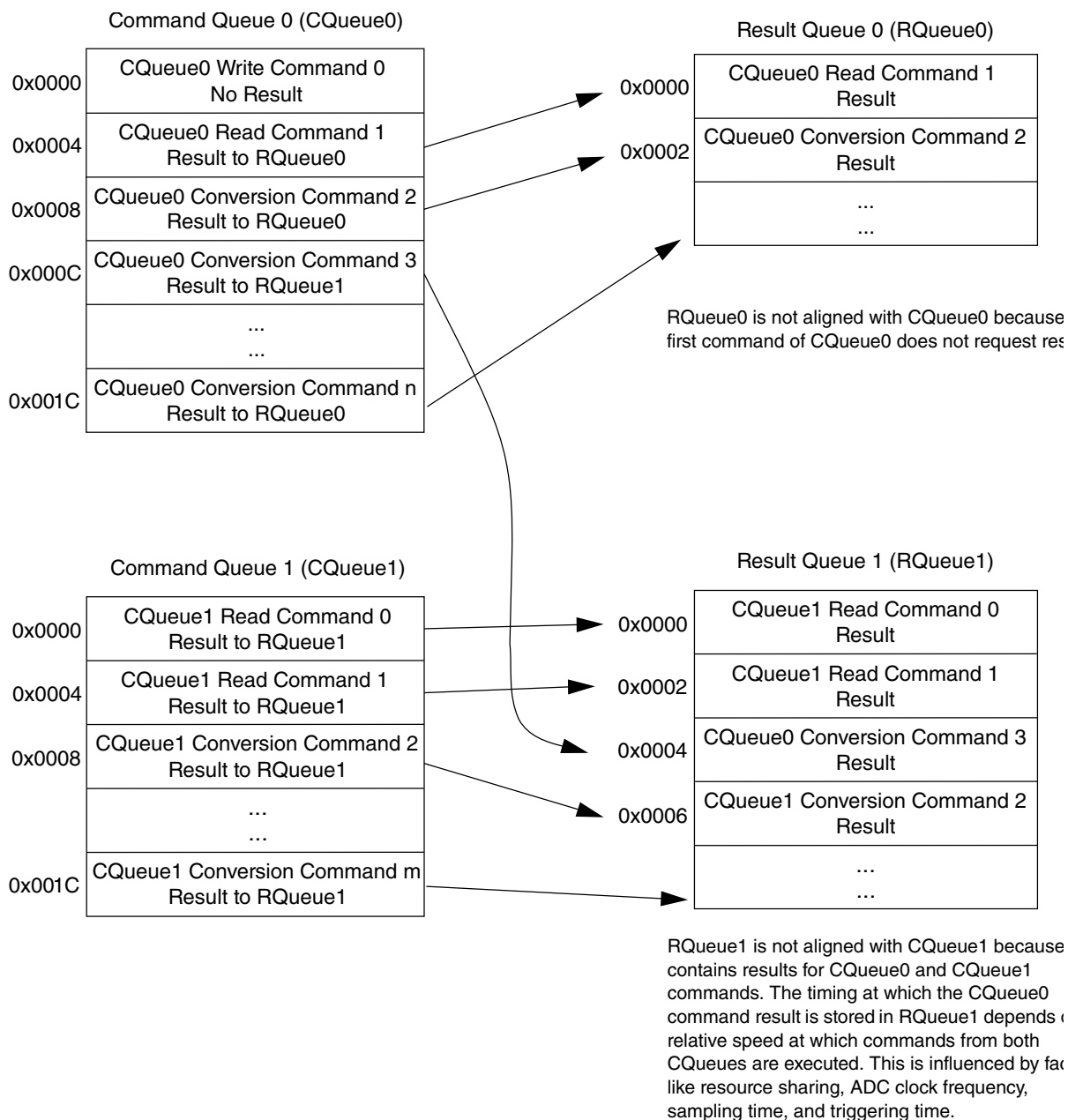


Figure 29-157. EQADC Command and Result Queues

## 29.7.6 ADC Result Calibration

The ADC result calibration process consists of two steps: determining the gain and offset calibration constants, and calibrating the raw results generated by the on-chip ADCs by solving the following equation discussed in [ADC calibration feature](#).

$$\text{CAL\_RES} = \text{GCC} * \text{RAW\_RES} + \text{OCC} + 2$$

**Figure 29-158. Equation for CAL\_RES**

The calibration constants GCC and OCC can be calculated from equation [Figure 29-159](#), [Figure 29-160](#), and [Figure 29-161](#) provided that two pairs of expected (CAL\_RES) and measured (RAW\_RES) result values are available for two different input voltages. Most likely calibration points to be used are 25% VREF<sup>7</sup> and 75% VREF since they are far apart but not too close to the end points of the full input voltage range. This allows for calculations of more representative calibration constants. The EQADC provides these voltages via channel numbers 43 and 44. The raw, uncalibrated results for these input voltages are obtained by converting these channels with conversion commands that have the CAL bit negated.

The transfer equations for when sampling these reference voltages are:

$$\text{CAL\_RES}_{75\% \text{VREF}} = \text{GCC} * \text{RAW\_RES}_{75\% \text{VREF}} + \text{OCC} + 2;$$

$$\text{CAL\_RES}_{25\% \text{VREF}} = \text{GCC} * \text{RAW\_RES}_{25\% \text{VREF}} + \text{OCC} + 2;$$

Thus;

$$\text{GCC} = (\text{CAL\_RES}_{75\% \text{VREF}} - \text{CAL\_RES}_{25\% \text{VREF}}) / (\text{RAW\_RES}_{75\% \text{VREF}} - \text{RAW\_RES}_{25\% \text{VREF}})$$

**Figure 29-159. Equation for GCC**

$$\text{OCC} = \text{CAL\_RES}_{75\% \text{VREF}} - \text{GCC} * \text{RAW\_RES}_{75\% \text{VREF}} - 2$$

**Figure 29-160. Equation for OCC, form 1**

or

$$\text{OCC} = \text{CAL\_RES}_{25\% \text{VREF}} - \text{GCC} * \text{RAW\_RES}_{25\% \text{VREF}} - 2$$

**Figure 29-161. Equation for OCC, form 2**

After being calculated, the GCC and OCC values must be written to ADC registers: [ADC0/1 Gain Calibration Constant Registers \(ADC0\\_GCCR and ADC1\\_GCCR\)](#) and [ADC0/1 Offset Calibration Constant Registers \(ADC0\\_OCCR and ADC1\\_OCCR\)](#) using write configuration commands.

7. VREF=VRH-VRL



The EQADC will automatically calibrate the results, according to equation [Figure 29-158](#), of every conversion command that has its CAL bit asserted using the GCC and OCC values stored in the ADC calibration registers.

### 29.7.6.1 MAC configuration procedure

The following steps illustrate how to configure the calibration hardware, namely, determining the values of the gain and offset calibration constants, and the writing of these constants to the calibration registers. The procedure below should be performed for ADC0 and for ADC1.

1. Convert channel 44 with a command that has its CAL bit negated and obtain the raw, uncalibrated result for 25%VREF (RAW\_RES<sub>25%VREF</sub>).
2. Convert channel 43 with a command that has its CAL bit negated and obtain the raw, uncalibrated result for 75%VREF (RAW\_RES<sub>75%VREF</sub>).
3. Since the expected values for the conversion of these voltages are known (CAL\_RES<sub>25%VREF</sub> and CAL\_RES<sub>75%VREF</sub>), GCC and OCC values can be calculated from [Figure 29-159](#) and [Figure 29-160](#) using these values, and the ones determined in steps 1 and 2.
4. Reformat GCC and OCC to the proper data formats as specified in [MAC unit and operand data format](#). GCC is an unsigned 15-bit fixed point value and OCC is a signed 14-bit value.
5. Write GCC value to [ADC0/1 Gain Calibration Constant Registers \(ADC0\\_GCCR and ADC1\\_GCCR\)](#), and OCC value to [ADC0/1 Offset Calibration Constant Registers \(ADC0\\_OCCR and ADC1\\_OCCR\)](#) using write configuration commands.

### 29.7.6.2 Example

The raw results obtained when sampling reference voltages 25%VREF and 75%VREF were, respectively, 3798 and 11592. The results that should have been obtained from the conversion of these reference voltages are, respectively, 4096 and 12288. Therefore, using equations [Figure 29-159](#) and [Figure 29-160](#), the gain and offset calibration constants are:

$$GCC = (12288 - 4096) / (11592 - 3798) = 1.05106492 \rightarrow 1.05102539 = 0x4344$$

$$OCC = 12288 - 1.05106492 * 11592 - 2 = 102.06 \rightarrow 102 = 0x0066$$



Table 29-145 shows, for this particular case, examples of how the result values change according to GCC and OCC when result calibration is executed (CAL=1) and when it is not (CAL=0).

**Table 29-145. Calibration example**

| Input voltage | Raw result (CAL=0) |         | Calibrated result (CAL=1) |           |
|---------------|--------------------|---------|---------------------------|-----------|
|               | Hexadecimal        | Decimal | Hexadecimal               | Decimal   |
| 25% VREF      | 0x0ED6             | 3798    | 0x1000                    | 4095.794  |
| 75% VREF      | 0x2D48             | 11592   | 0x3000                    | 12287.486 |

### 29.7.6.3 Quantization error reduction during calibration

Figure 29-162 shows how the ADC transfer curve changes due to the addition of two to the MAC output during the calibration - see MAC output equation at [Overview](#) . The maximum absolute quantization error is reduced by half leading to an increase in accuracy.

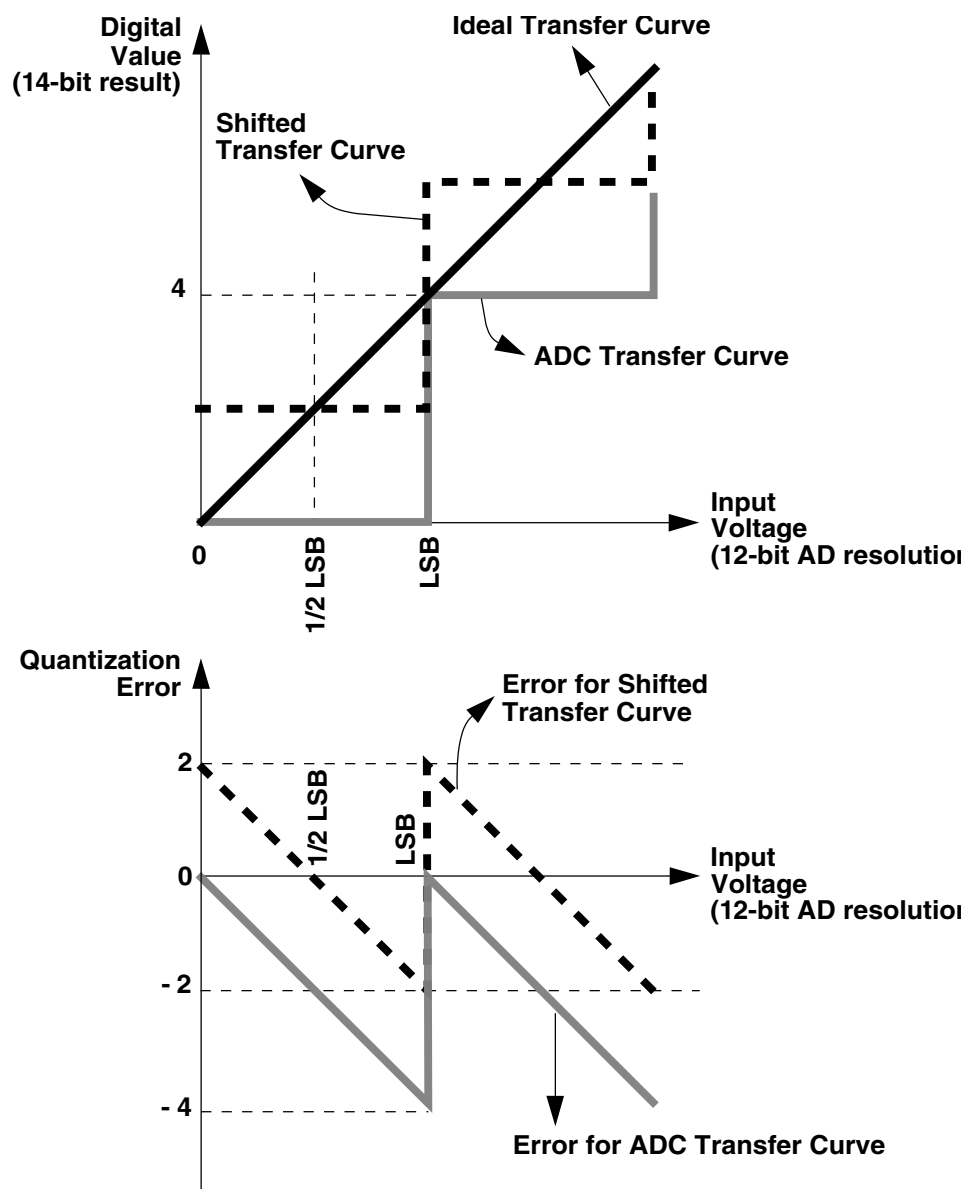


Figure 29-162. Quantization error reduction during calibration

### 29.7.7 EQADC versus QADC

This section describes how the EQADC upgrades the QADC functionality. The section also provides a comparison between the EQADC and QADC in terms of their functionality. This section targets the users familiar with terminology in QADC. [Figure 29-163](#) is an overview of a QADC. [Figure 29-164](#) is an overview of the EQADC system.

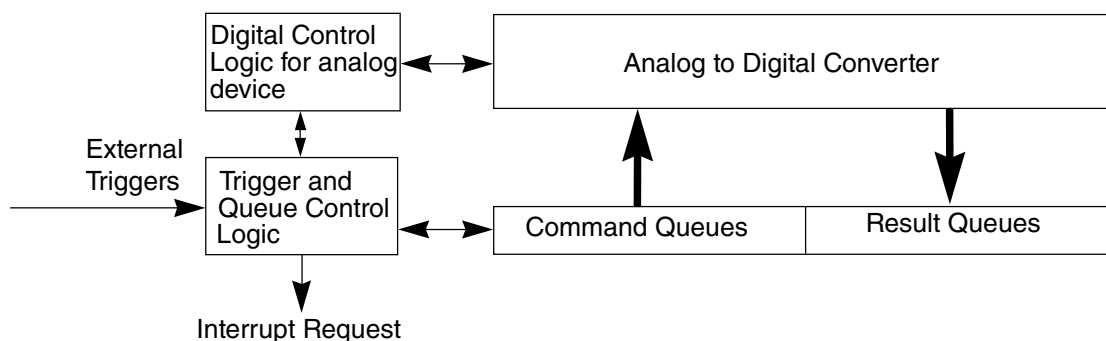


Figure 29-163. QADC overview

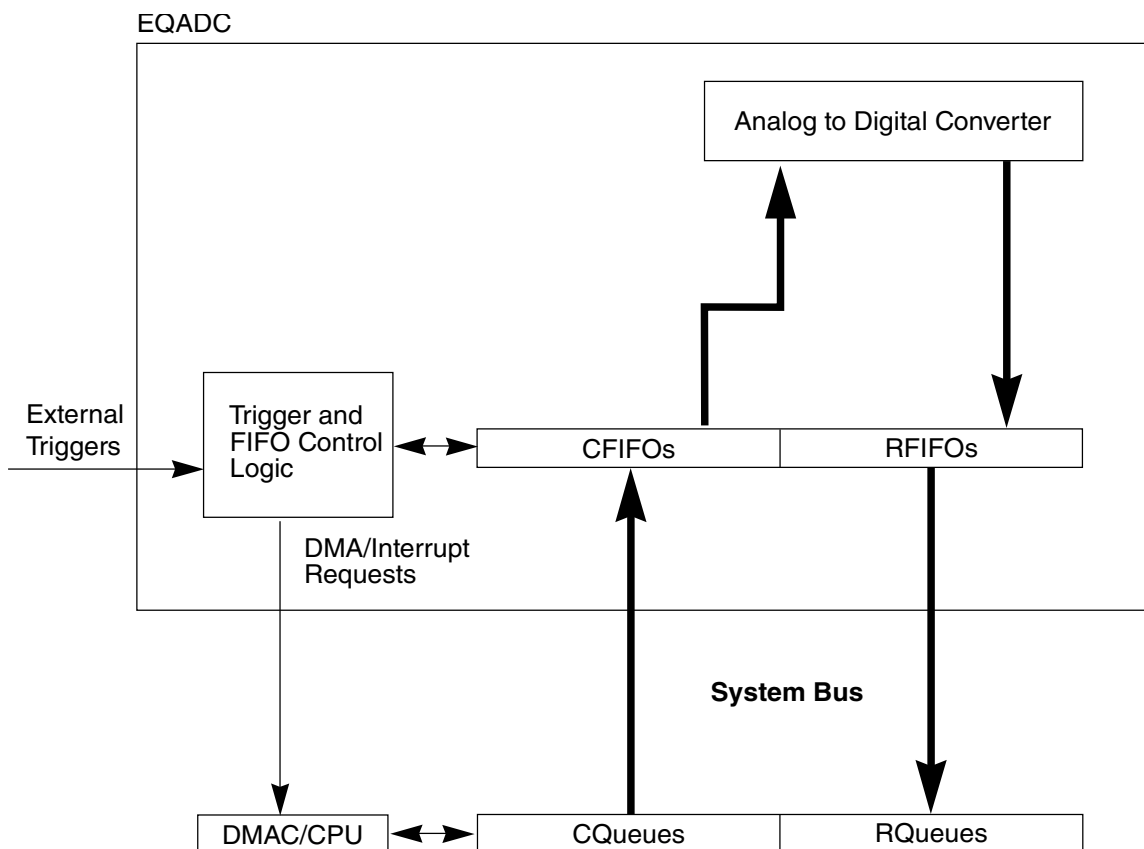


Figure 29-164. EQADC system overview

The EQADC system consists of queues in RAM, an external device, the EQADC, and on-chip ADCs. As compared with the QADC, the EQADC system requires extra hardware.

- A DMA or an MCU is required to move data between the EQADC's FIFOs and Queues in the system memory.

Since there are only FIFOs inside the EQADC, much of the terminology or use of the register names, register contents, and signals of the EQADC involve "FIFO" instead of "Queue". These register names, register contents, and signals are functionally equivalent to the "Queue" counterparts in the QADC. [Table 29-146](#) lists how the EQADC register, register contents, and signals are related to QADC.

**Table 29-146. Terminology comparison between QADC and EQADC**

| QADC terminology                     | EQADC terminology                                                 | Function                                                                                                                                                                                                                                                                                                                                                                                                                |
|--------------------------------------|-------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CCW                                  | Command Message                                                   | In the QADC, the hardware only executes conversion command words.<br><br>In the EQADC, not all commands are conversion commands; some are configuration commands.                                                                                                                                                                                                                                                       |
| Queue Trigger                        | CFIFO Trigger                                                     | In the QADC, a trigger event is required to start the execution of a queue.<br><br>In the EQADC, a trigger event is required to start command transfers from a CFIFO. When a CFIFO is TRIGGERED and transferring, commands are continuously moved from CQueues to CFIFOs. Thus, the trigger event initiates the "execution of a queue" indirectly.                                                                      |
| Current Word Pointer Queue x (CWPQx) | Counter Value of Commands Transferred from Command FIFOx (TC_CFx) | In the QADC, CWPQx allows the last executed command on queue x to be determined.<br><br>In the EQADC, the TC_CFx value allows the last transferred command on CQueue x to be determined.                                                                                                                                                                                                                                |
| Queue Pause Bit (P)                  | CFIFO Pause Bit                                                   | In the QADC, detecting a pause bit in the CCW will pause the queue execution.<br><br>In the EQADC, detecting a pause bit in the Command will pause command transfers from a CFIFO.                                                                                                                                                                                                                                      |
| Queue Operation Mode (MQx)           | CFIFO Operation Mode (MODEx)                                      | The EQADC supports all queue operation modes in the QADC except operation modes related to a periodic timer. A timer elsewhere in the system can provide the same functionality if it is connected to ETRIGx.                                                                                                                                                                                                           |
| Queue Status (QS)                    | CFIFO Status (CFSx)                                               | In the QADC, the Queue Status is read to check whether a queue is idle, active, paused, suspended, or trigger pending.<br><br>In the EQADC, the CFIFO Status is read to check whether a queue is IDLE, WAITING FOR TRIGGER (idle or paused in QADC), or TRIGGERED (suspended or trigger pending in QADC). What CFIFO is currently "active" can be determined by reading the LCFTCBn field on the EQADC_CFSSR registers. |

The EQADC and QADC also have similar procedures for the configuration or execution of applications. [Table 29-147](#) shows the steps required for the QADC versus the steps required for the EQADC system.

**Table 29-147. Usage comparison between QADC and EQADC system**

| Procedure                    | QADC                                                      | EQADC System                                        |
|------------------------------|-----------------------------------------------------------|-----------------------------------------------------|
| Analog Control Configuration | Configure analog device by writing to the QADC registers. | Program configuration commands into command queues. |
| Prepare Scan Sequence        | Program scan commands into command queues.                | Program scan commands into command queues.          |
| Queue Control Configuration  | Write to the QADC Control Registers.                      | Write to the EQADC Control Registers.               |

*Table continues on the next page...*

**Table 29-147. Usage comparison between QADC and EQADC system (continued)**

| Procedure                                   | QADC                                                                  | EQADC System                                                                         |
|---------------------------------------------|-----------------------------------------------------------------------|--------------------------------------------------------------------------------------|
| Data Transferred between Queues and Buffers | Not Required.                                                         | Program the DMAC or the CPU to handle the data transfer.                             |
| Queue Execution                             | Require Software or External Trigger events to start queue execution. | Require Software or External Trigger events to start command transfers from a CFIFO. |



## Chapter 30

# Temperature Sensor (TSENS)

### 30.1 Chip-specific Temperature Sensor information

This chip has two instances of the Temperature Sensor.

#### 30.1.1 Temperature Sensor programming control

The control and status registers for the Temperature Sensor are implemented in the PMC digital interface.

#### 30.1.2 Accessing TSC parameters

The Temperature Sensor Calibration (TSC) parameters, including TSCA and TSCB, are used to calculate device temperature.

See the UTEST flash memory map for the specific locations of the TSC parameter values.

#### 30.1.3 Temperature name alignment

The Temperature Sensor modules and Power Management Controller (PMC) module use different terminology to refer to the same temperature points. The following table aligns the terminology.

**Table 30-1. Temperature Sensor and PMC temperature terminology**

| Temperature | Temperature Sensor name | PMC name      |
|-------------|-------------------------|---------------|
| -40°C       | Low temperature         | Temperature 0 |
| 150°C       | High temperature        | Temperature 2 |

## 30.2 Introduction

The device includes two on-chip temperature sensors. The temperature sensor module monitors device temperature and delivers one analog output signal and two digital output signals each (under- and over-temperature flags). The analog output consists of a voltage signal directly proportional to the internal junction temperature. The analog output is connected to an input channel of an ADC on the device (see the ADC details for the channel number). The internal junction temperature must be calculated by software based on the sampled temperature sensor voltage; sampled bandgap reference voltage, which comes from another module; and calibration parameter values stored in internal flash memory. The two digital outputs, connected to the PMC module, are used to signal under- and over-temperature operating conditions.

The digital outputs signal when the junction temperature drifts below the low temperature threshold or above the high temperature threshold. These signals notify the device to take action to appropriately adjust the device temperature in response to an out-of-specification low or high temperature operating condition. Calibration parameter values associated with the temperature threshold detection feature are determined and stored in internal flash memory during production testing at the factory.

## 30.3 Functional Description

The temperature sensor generates one analog output voltage,  $V_{TSENS}(T)$ , which is proportional to the absolute current junction temperature of the device, and two digital outputs that signal whether the junction temperature has reached either a pre-set low temperature threshold or a pre-set high temperature threshold.

An on-chip ADC module is used to convert the analog output voltage, as well as the bandgap reference voltage, into a digital representation. These values, along with parameter values stored in on-chip flash memory, are used by software to calculate the device junction temperature.



### 30.3.1 Temperature threshold detection (digital output generation)

Temperature threshold trimming values are adjusted through calibration during factory test and stored in on-chip flash memory. While enabled, if a new trimming value is loaded into the Temperature Sensor module, it may take up to 1 microsecond for the digital outputs to be updated to reflect the new trimming condition.

When the temperature threshold detection feature is enabled, the temperature sensor monitors the internal junction temperature of the chip and asserts a signal if any of the following temperature thresholds are crossed.

- The low temperature digital output signals if the junction temperature falls below the low temperature threshold ( $-40^{\circ}\text{C}$ ).
- The high temperature digital output signals if the junction temperature rises above the high temperature threshold ( $150^{\circ}\text{C}$ ).

The threshold detection circuit is calibrated with respect to the factory-test temperatures: the low temperature threshold is trimmed to match the cold insertion test temperature (nominally equal to  $-40^{\circ}\text{C}$ ), and the high temperature threshold is trimmed to match the hot insertion test temperature (nominally equal to  $150^{\circ}\text{C}$ ). In other words, the thresholds are not trimmed to absolute values of temperature but to the specific temperatures for which the device is validated during factory test.

Hysteresis is applied whenever outputs transition to a logic level low in order to eliminate spurious toggling. Digital output behavior is illustrated on [Figure 30-1](#).

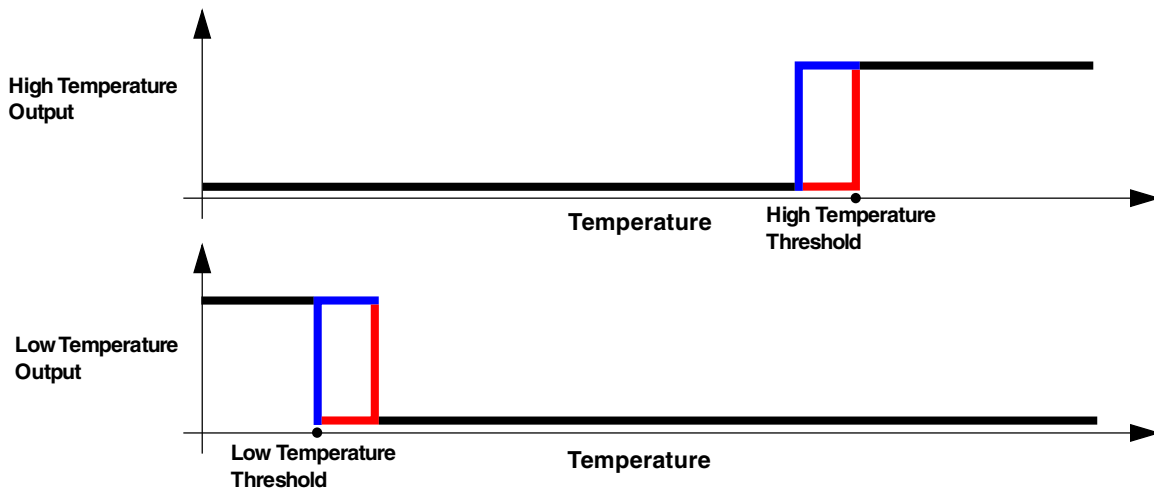


**Figure 30-1. Digital output behavior with temperature**

### 30.3.2 Linear temperature sensor (analog output generation)

When analog output generation is enabled, the temperature sensor outputs a voltage proportional to the internal junction temperature of the chip. This analog voltage signal is converted into a digital code by an on-chip ADC. The temperature value is obtained from a linear voltage-temperature relation with coefficients adjusted by calibration parameters extracted during factory test and programmed into flash memory.

## 30.4 Temperature formula

The system chain that translates device junction temperature into a digital variable is composed of the temperature sensor, a bandgap reference voltage source and the on-chip ADC. Both analog output voltages of the temperature sensor and the bandgap reference voltage source must be converted by the ADC into digital codes to obtain the device junction temperature. The temperature formula shown in [Figure 30-2](#) is used to calculate internal device junction temperature.

$$T = (232 + TSCA \times 2^{-6}) \times \frac{V_{TSENS\_CODE}(T)}{V_{BG\_CODE}(T)} - (273 + TSCB \times 2^{-4}) [^{\circ}\text{C}]$$

Where:

$$TSCA = \left( \frac{T_{HIGH} - T_{LOW}}{V_{TSENS\_CODE}(T_{HIGH}) - V_{TSENS\_CODE}(T_{LOW})} \times V_{BG\_CODE}(T_{FINAL}) - 232 \right) \times 2^6$$

$$TSCB = \left( \frac{T_{HIGH} - T_{LOW}}{V_{TSENS\_CODE}(T_{HIGH}) - V_{TSENS\_CODE}(T_{LOW})} \times V_{TSENS\_CODE}(T_{LOW}) - T_{LOW} - 273 \right) \times 2^4$$

$$V_{TSENS\_CODE}(T_{LOW}) = \frac{V_{TSENS}(T_{LOW})}{V_{ref0}} \times 2^{12}$$

$$V_{TSENS\_CODE}(T_{HIGH}) = \frac{V_{TSENS}(T_{HIGH})}{V_{ref0}} \times 2^{12}$$

$$V_{BG\_CODE}(T_{FINAL}) = \frac{V_{BG}(T_{FINAL})}{V_{ref0}} \times 2^{12}$$

$$V_{BG\_CODE}(T) = \frac{V_{BG}(T)}{V_{ref}} \times 2^{12}$$

$$V_{TSENS\_CODE}(T) = \frac{V_{TSENS}(T)}{V_{ref}} \times 2^{12}$$

**Notes:**

- TSCA and TSCB are 16-bit signed integer constants stored in flash memory during factory test and calibration
- $V_{TSENS}(T)$  is the temperature sensor output sampled by the ADC
- $V_{BG}(T)$  is the bandgap reference voltage sampled by the ADC at temperature T
- $V_{ref}$  is the ADC reference voltage in normal usage of the device
- $V_{ref0}$  is the ADC reference voltage during factory calibration
- $T_{LOW}$  is the low temperature factory calibration temperature
- $T_{HIGH}$  is the hot temperature factory calibration temperature
- $T_{FINAL}$  is the final insertion factory calibration temperature

**Figure 30-2. Temperature formula**

Parameters TSCA and TSCB are 16-bit signed integer constants stored during factory test and calibration that need to be used in the temperature formula to calculate device junction temperature. The values for TSCA and TSCB translate into positive or negative shifts of the linear relation coefficients that compensate for manufacturing process variation. [Figure 30-2](#) shows how parameters TSCA and TSCB are calculated in terms of the actual variables measured during factory calibration.

## 30.5 Calculating device temperature

Software must perform the following steps in order to calculate device temperature.

#### Calculating device temperature

1. Read the bandgap reference voltage  $V_{\text{TSENS\_CODE}}(T)$  using the on-chip ADC module.
2. Read the temperature sensor output  $V_{\text{BG\_CODE}}(T)$  using the on-chip ADC module.
3. Retrieve parameters TSCA and TSCB from flash memory.
4. Calculate T according to the formula shown in [Figure 30-2](#).

#### Note

Temperature Formula may be calculated using signed 32-bit (integer) operations showing negligible precision loss.

# Chapter 31

## System Timer Module (STM)

### 31.1 Chip-specific STM information

This chip has one instance of the STM.

### 31.2 Introduction

This section provides an overview, list of features, and modes of operation for the STM.

#### 31.2.1 Overview

The System Timer Module (STM) is a 32-bit timer designed to support commonly required system and application software timing functions. The STM includes a 32-bit up counter and four 32-bit compare channels with a separate interrupt source for each channel. The counter is driven by the system clock divided by an 8-bit prescale value (1 to 256).

#### 31.2.2 Features

The STM has the following features:

- One 32-bit up counter with 8-bit prescaler
- Four 32-bit compare channels
- Independent interrupt source for each channel
- Counter can be stopped in debug mode

### 31.2.3 Modes of operation

The STM supports two device modes of operation: normal and debug. When the STM is enabled in normal mode, its counter runs continuously. In debug mode, operation of the counter is controlled by the FRZ bit in the STM\_CR register. If the FRZ bit is set, the counter is stopped in debug mode, otherwise it continues to run.

## 31.3 External signal description

The STM does not have any external interface signals.

## 31.4 Memory map and registers

The STM programming model includes a group of 32-bit registers—a module control register, a counter value register, and three registers for each channel.

The STM registers can only be accessed using 32-bit (word) accesses. Attempted references using a different size or to a reserved address generates a bus error termination.

**STM memory map**

| Address offset (hex) | Register name                             | Width (in bits) | Access | Reset value | Section/page                |
|----------------------|-------------------------------------------|-----------------|--------|-------------|-----------------------------|
| 0                    | STM Control Register (STM_CR)             | 32              | R/W    | 0000_0000h  | <a href="#">31.4.1/1155</a> |
| 4                    | STM Count Register (STM_CNT)              | 32              | R/W    | 0000_0000h  | <a href="#">31.4.2/1156</a> |
| 10                   | STM Channel Control Register (STM_CCR0)   | 32              | R/W    | 0000_0000h  | <a href="#">31.4.3/1156</a> |
| 14                   | STM Channel Interrupt Register (STM_CIR0) | 32              | w1c    | 0000_0000h  | <a href="#">31.4.4/1157</a> |
| 18                   | STM Channel Compare Register (STM_CMP0)   | 32              | R/W    | 0000_0000h  | <a href="#">31.4.5/1157</a> |
| 20                   | STM Channel Control Register (STM_CCR1)   | 32              | R/W    | 0000_0000h  | <a href="#">31.4.3/1156</a> |
| 24                   | STM Channel Interrupt Register (STM_CIR1) | 32              | w1c    | 0000_0000h  | <a href="#">31.4.4/1157</a> |
| 28                   | STM Channel Compare Register (STM_CMP1)   | 32              | R/W    | 0000_0000h  | <a href="#">31.4.5/1157</a> |
| 30                   | STM Channel Control Register (STM_CCR2)   | 32              | R/W    | 0000_0000h  | <a href="#">31.4.3/1156</a> |
| 34                   | STM Channel Interrupt Register (STM_CIR2) | 32              | w1c    | 0000_0000h  | <a href="#">31.4.4/1157</a> |
| 38                   | STM Channel Compare Register (STM_CMP2)   | 32              | R/W    | 0000_0000h  | <a href="#">31.4.5/1157</a> |
| 40                   | STM Channel Control Register (STM_CCR3)   | 32              | R/W    | 0000_0000h  | <a href="#">31.4.3/1156</a> |
| 44                   | STM Channel Interrupt Register (STM_CIR3) | 32              | w1c    | 0000_0000h  | <a href="#">31.4.4/1157</a> |
| 48                   | STM Channel Compare Register (STM_CMP3)   | 32              | R/W    | 0000_0000h  | <a href="#">31.4.5/1157</a> |

### 31.4.1 STM Control Register (STM\_CR)

The STM Control Register includes the prescale value, freeze control and timer enable bits.

Address: 0h base + 0h offset = 0h

|       |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0   | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| Bit   | 16  | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | CPS |    |    |    |    |    |    |    | 0  |    |    |    |    |    |    |    |
| W     |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

#### STM\_CR field descriptions

| Field             | Description                                                                                                                                                                                                                                                                                          |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                              |
| 16–23<br>CPS      | Counter Prescaler.<br>Selects the clock divide value for the prescaler (1 - 256).<br>0x00 Divide system clock by 1<br>0x01 Divide system clock by 2<br>-----<br>0xFF Divide system clock by 256                                                                                                      |
| 24–29<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                              |
| 30<br>FRZ         | Freeze.<br>Allows the timer counter to be stopped when the device enters debug mode.<br><b>NOTE:</b> When the MCU enters debug mode, the STM is notified and uses the FRZ bit to determine counter mode.<br>0 STM counter continues to run in debug mode.<br>1 STM counter is stopped in debug mode. |
| 31<br>TEN         | Timer counter Enabled.<br>0 Counter is disabled.<br>1 Counter is enabled.                                                                                                                                                                                                                            |

## 31.4.2 STM Count Register (STM\_CNT)

The STM Count Register holds the timer count value.

Address: 0h base + 4h offset = 4h

|       |     |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-----|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | CNT |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |     |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

### STM\_CNT field descriptions

| Field       | Description                                                                                                                                                      |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>CNT | Timer count value used as the time base for all channels.<br>When enabled, the counter increments at the rate of the system clock divided by the prescale value. |

## 31.4.3 STM Channel Control Register (STM\_CCRn)

The STM Channel Control Register (STM\_CCRn) has the enable bit for channel n of the timer.

Address: 0h base + 10h offset + (16d × i), where i=0d to 3d

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |     |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | CEN |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |     |

### STM\_CCRn field descriptions

| Field            | Description                                                                             |
|------------------|-----------------------------------------------------------------------------------------|
| 0–30<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 31<br>CEN        | Channel Enable<br>0 The channel is disabled.<br>1 The channel is enabled.               |



### 31.4.4 STM Channel Interrupt Register (STM\_CIR<sub>n</sub>)

The STM Channel Interrupt Register has the interrupt flag for channel *n* of the timer.

Address: 0h base + 14h offset + (16d × *i*), where *i*=0d to 3d

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31  |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    | CIF |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | w1c |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   |

#### STM\_CIR<sub>n</sub> field descriptions

| Field            | Description                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------------------|
| 0–30<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                     |
| 31<br>CIF        | Channel Interrupt Flag<br><br>0 No interrupt request.<br>1 Interrupt request due to a match on the channel. |

### 31.4.5 STM Channel Compare Register (STM\_CMP<sub>n</sub>)

The STM channel compare register (STM\_CMP<sub>n</sub>) holds the compare value for channel *n*.

Address: 0h base + 18h offset + (16d × *i*), where *i*=0d to 3d

|       |                |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0              | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | <div>CMP</div> |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |                |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0              | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

#### STM\_CMP<sub>n</sub> field descriptions

| Field       | Description                                                                                                                                                                                                                                          |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>CMP | Compare value for channel <i>n</i> .<br><br>If the STM_CCR <sub>n</sub> [CEN] bit is set and the STM_CMP <sub>n</sub> register matches the STM_CNT register, a channel interrupt request is generated and the STM_CIR <sub>n</sub> [CIF] bit is set. |

STM\_CMPn field descriptions (continued)

| Field | Description |
|-------|-------------|
|-------|-------------|

31.5 Functional description

The STM has one 32-bit up counter (STM\_CNT) that is used as the time base for all channels. When enabled, the counter increments at the system clock frequency divided by a prescale value. The STM\_CR[CPS] field sets the divider to any value in the range from 1 to 256. The counter is enabled with the STM\_CR[TEN] bit. When enabled in normal mode, the counter continuously increments. When enabled in debug mode, the counter operation is controlled by the STM\_CR[FRZ] bit. When the STM\_CR[FRZ] bit is set, the counter is stopped in debug mode; otherwise, it continues to run in debug mode. The counter rolls over at 0xFFFF\_FFFF to 0x0000\_0000 with no restrictions at this boundary.

The STM has four identical compare channels. Each channel includes a channel control register (STM\_CCRn), a channel interrupt register (STM\_CIRn), and a channel compare register (STM\_CMPn). The channel is enabled by setting the STM\_CCRn[CEN] bit. When enabled, the channel will set the STM\_CIRn[CIF] bit and generate an interrupt request when the channel compare register matches the timer counter. The interrupt request is cleared by writing 1 to the STM\_CIRn[CIF] bit. A write of 0 to the STM\_CIRn[CIF] bit has no effect.

NOTE

The STM counter does not advance when the system clock is stopped.

# Chapter 32

## Software Watchdog Timer (SWT)

### 32.1 Chip-specific SWT information

This chip has two instances of the SWT module: SWT\_A and SWT\_B.

#### 32.1.1 SWT register reset values

The following table identifies chip-specific reset values of SWT registers.

**Table 32-1. Chip-specific SWT register reset values**

| Register | SWT_A                   | SWT_B      |
|----------|-------------------------|------------|
| CR       | FF00_010Bh <sup>1</sup> | FF00_010Ah |
| TO       | 0005_FCD0h              | 0005_FCD0h |

1. This value is present after a hardware reset and before the BAM executes. The BAM code (RCHW) might change it before application code executes. See [Reset Configuration Half Word \(RCHW\)](#) for more information.

#### 32.1.2 SWT time-out events and invalid accesses: configuring reset and interrupt requests

To configure the results of SWT time-out events and invalid accesses for an SWT that is enabled (SWT\_CR[WEN] is 1):

- Program these [SWT Control Register \(SWT\\_CR\)](#) fields:

- **ITR** determines when the outcome of a time-out event is a reset request or an interrupt request.
- **RIA** determines when the outcome of an invalid access is a reset request or a bus error.
- Program FCCU reactions to the non-critical faults (NCFs) that are associated with these reset and interrupt requests.

See **FCCU fault inputs** for the default reaction to each NCF.

### CAUTION

For any NCF that receives an SWT reset request: If the default FCCU reaction is no reset but the intended application requires a reset, configure the FCCU's reaction to be a reset.

The following table shows the NCFs to which the SWT generates reset and interrupt requests for time-out events.

**Table 32-2. Routing of SWT reset and interrupt requests for time-out events**

| SWT instance | SWT_CR[ITR] is 0                        | SWT_CR[ITR] is 1                                |                                                        |
|--------------|-----------------------------------------|-------------------------------------------------|--------------------------------------------------------|
|              | Any time-out generates reset request to | Initial time-out generates interrupt request to | Second consecutive time-out generates reset request to |
| SWT_A        | NCF[14]                                 | IRQ[8] and NCF[16]                              | NCF[14]                                                |
| SWT_B        | NCF[15]                                 | IRQ[501]                                        | NCF[15]                                                |

The following table shows the NCFs to which the SWT can generate reset requests for invalid accesses.

**Table 32-3. Routing of SWT reset requests for invalid accesses**

| SWT instance | SWT_CR[RIA] is 0                   | SWT_CR[RIA] is 1                                  |
|--------------|------------------------------------|---------------------------------------------------|
| SWT_A        | Invalid access generates bus error | Invalid access generates reset request to NCF[14] |
| SWT_B        | Invalid access generates bus error | Invalid access generates reset request to NCF[15] |

## 32.2 Introduction

This section provides an overview, list of features, and modes of operation for the SWT.

### 32.2.1 Overview

The Software Watchdog Timer (SWT) is a peripheral module that can prevent system lockup in situations such as software getting trapped in a loop or if a bus transaction fails to terminate. When enabled, the SWT requires periodic execution of a watchdog servicing operation. The servicing operation resets the timer to a specified time-out period. If this servicing action does not occur before the timer expires, the SWT generates an interrupt or hardware reset request. The SWT can be configured to generate a reset request or interrupt on an initial time-out. The SWT always generates a reset request on a second consecutive time-out.

### 32.2.2 Features

The SWT has the following features:

- 32-bit time-out register to set the time-out period
- Programmable selection of system or oscillator clock, for timer operation
- Programmable selection of window mode or regular servicing
- Programmable selection of reset request or interrupt on an initial time-out
- Programmable selection of the servicing mode
- Master access protection
- Hard and soft configuration lock bits
- Reset configuration inputs allow timer to be enabled out of reset

### 32.2.3 Modes of operation

The SWT supports these device modes of operation:

- Normal
- Debug
- Stop

When the SWT is enabled in normal mode, its counter runs continuously. In debug mode, operation of the counter is controlled by the FRZ bit in the SWT\_CR. If the FRZ bit is set, the counter is stopped in debug mode; otherwise it continues to run. In stop mode, operation of the counter is controlled by the STP bit in the SWT\_CR. If the STP bit is set, the counter is stopped in stop mode; otherwise it continues to run.

### 32.3 External signal description

The SWT module does not have any external interface signals.

### 32.4 Memory Map and Registers

The SWT programming model has seven 32-bit registers. The programming model can be accessed using only 32-bit (word) accesses.

Any of the following attempted accesses are invalid:

- References using non-32-bit access sizes
- Writes to read-only registers
- Writes of incorrect values to SWT\_SR when the SWT is enabled
- Accesses to reserved addresses
- Accesses by masters without permission

For an enabled SWT, an invalid access generates a reset request if SWT\_CR[RIA] is 1. An invalid access generates a bus error if SWT\_CR[RIA] is 0.

If either HLK or SLK in SWT\_CR is 1, then the SWT\_CR, SWT\_TO, SWT\_WN, and SWT\_SK registers are read-only.

**SWT memory map**

| Address offset (hex) | Register name                        | Width (in bits) | Access | Reset value                 | Section/ page               |
|----------------------|--------------------------------------|-----------------|--------|-----------------------------|-----------------------------|
| 0                    | SWT Control Register (SWT_CR)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">32.4.1/1163</a> |
| 4                    | SWT Interrupt Register (SWT_IR)      | 32              | R/W    | 0000_0000h                  | <a href="#">32.4.2/1166</a> |
| 8                    | SWT Time-out Register (SWT_TO)       | 32              | R/W    | <a href="#">See section</a> | <a href="#">32.4.3/1167</a> |
| C                    | SWT Window Register (SWT_WN)         | 32              | R/W    | 0000_0000h                  | <a href="#">32.4.4/1167</a> |
| 10                   | SWT Service Register (SWT_SR)        | 32              | W      | 0000_0000h                  | <a href="#">32.4.5/1168</a> |
| 14                   | SWT Counter Output Register (SWT_CO) | 32              | R      | 0000_0000h                  | <a href="#">32.4.6/1168</a> |
| 18                   | SWT Service Key Register (SWT_SK)    | 32              | R/W    | 0000_0000h                  | <a href="#">32.4.7/1169</a> |

### 32.4.1 SWT Control Register (SWT\_CR)

#### NOTE

The reset value of this register is implementation specific. See the chip-specific SWT information.

The SWT\_CR contains fields for configuring and controlling the SWT.

This register is read-only if either SWT\_CR[HLK] or SWT\_CR[SLK] is 1.

Address: 0h base + 0h offset = 0h

|       |      |      |      |      |      |      |      |      |     |     |     |     |     |     |     |     |
|-------|------|------|------|------|------|------|------|------|-----|-----|-----|-----|-----|-----|-----|-----|
| Bit   | 0    | 1    | 2    | 3    | 4    | 5    | 6    | 7    | 8   | 9   | 10  | 11  | 12  | 13  | 14  | 15  |
| R     | MAP0 | MAP1 | MAP2 | MAP3 | MAP4 | MAP5 | MAP6 | MAP7 | 0   |     |     |     |     |     |     |     |
| W     |      |      |      |      |      |      |      |      |     |     |     |     |     |     |     |     |
| Reset | 0*   | 0*   | 0*   | 0*   | 0*   | 0*   | 0*   | 0*   | 0*  | 0*  | 0*  | 0*  | 0*  | 0*  | 0*  | 0*  |
| Bit   | 16   | 17   | 18   | 19   | 20   | 21   | 22   | 23   | 24  | 25  | 26  | 27  | 28  | 29  | 30  | 31  |
| R     | 0    |      |      |      |      | SMD  |      | RIA  | WND | ITR | HLK | SLK | CSL | STP | FRZ | WEN |
| W     |      |      |      |      |      |      |      |      |     |     |     |     |     |     |     |     |
| Reset | 0*   | 0*   | 0*   | 0*   | 0*   | 0*   | 0*   | 0*   | 0*  | 0*  | 0*  | 0*  | 0*  | 0*  | 0*  | 0*  |

\* Notes:

- 

#### SWT\_CR field descriptions

| Field     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>MAP0 | <p>Master Access Protection for Master 0. The platform bus master assignments are device specific.</p> <p>The number of this field corresponds to the logical ID of the bus master. MAP0 corresponds to logical IDs 0 and 8.</p> <p>For the masters with the logical IDs to which the MAP<math>n</math> fields correspond, see the chip-specific information for the Crossbar Switch.</p> <p>0 Access for the master is not enabled<br/>1 Access for the master is enabled</p> |
| 1<br>MAP1 | <p>Master Access Protection for Master 1. The platform bus master assignments are device specific.</p> <p>The number of this field corresponds to the logical ID of the bus master. MAP1 corresponds to logical IDs 1 and 9.</p> <p>For the masters with the logical IDs to which the MAP<math>n</math> fields correspond, see the chip-specific information for the Crossbar Switch.</p> <p>0 Access for the master is not enabled<br/>1 Access for the master is enabled</p> |

Table continues on the next page...

## SWT\_CR field descriptions (continued)

| Field     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2<br>MAP2 | <p>Master Access Protection for Master 2. The platform bus master assignments are device specific.</p> <p>The number of this field corresponds to the logical ID of the bus master. MAP2 corresponds to logical IDs 2 and 10.</p> <p>For the masters with the logical IDs to which the MAP<math>n</math> fields correspond, see the chip-specific information for the Crossbar Switch.</p> <p>0 Access for the master is not enabled<br/>1 Access for the master is enabled</p> |
| 3<br>MAP3 | <p>Master Access Protection for Master 3. The platform bus master assignments are device specific.</p> <p>The number of this field corresponds to the logical ID of the bus master. MAP3 corresponds to logical IDs 3 and 11.</p> <p>For the masters with the logical IDs to which the MAP<math>n</math> fields correspond, see the chip-specific information for the Crossbar Switch.</p> <p>0 Access for the master is not enabled<br/>1 Access for the master is enabled</p> |
| 4<br>MAP4 | <p>Master Access Protection for Master 4. The platform bus master assignments are device specific.</p> <p>The number of this field corresponds to the logical ID of the bus master. MAP4 corresponds to logical IDs 4 and 12.</p> <p>For the masters with the logical IDs to which the MAP<math>n</math> fields correspond, see the chip-specific information for the Crossbar Switch.</p> <p>0 Access for the master is not enabled<br/>1 Access for the master is enabled</p> |
| 5<br>MAP5 | <p>Master Access Protection for Master 5. The platform bus master assignments are device specific.</p> <p>The number of this field corresponds to the logical ID of the bus master. MAP5 corresponds to logical IDs 5 and 13.</p> <p>For the masters with the logical IDs to which the MAP<math>n</math> fields correspond, see the chip-specific information for the Crossbar Switch.</p> <p>0 Access for the master is not enabled<br/>1 Access for the master is enabled</p> |
| 6<br>MAP6 | <p>Master Access Protection for Master 6. The platform bus master assignments are device specific.</p> <p>The number of this field corresponds to the logical ID of the bus master. MAP6 corresponds to logical IDs 6 and 14.</p> <p>For the masters with the logical IDs to which the MAP<math>n</math> fields correspond, see the chip-specific information for the Crossbar Switch.</p> <p>0 Access for the master is not enabled<br/>1 Access for the master is enabled</p> |
| 7<br>MAP7 | <p>Master Access Protection for Master 7. The platform bus master assignments are device specific.</p> <p>The number of this field corresponds to the logical ID of the bus master. MAP7 corresponds to logical IDs 7 and 15.</p> <p>For the masters with the logical IDs to which the MAP<math>n</math> fields correspond, see the chip-specific information for the Crossbar Switch.</p>                                                                                      |

*Table continues on the next page...*



## SWT\_CR field descriptions (continued)

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                               |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                  | 0 Access for the master is not enabled<br>1 Access for the master is enabled                                                                                                                                                                                                                                                                                                                              |
| 8–20<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                   |
| 21–22<br>SMD     | Service Mode<br>00 Fixed Service Sequence: The watchdog is serviced by writing the fixed sequence 0xA602, 0xB480 to SWT_SR.<br>01 Keyed Service Sequence: The watchdog is serviced by writing two pseudorandom key values to SWT_SR.<br>10 Reserved—do not use. Writing a value can cause the SWT not to be serviced.<br>11 Reserved—do not use. Writing this value can cause the SWT not to be serviced. |
| 23<br>RIA        | Reset on Invalid Access<br><br><b>NOTE:</b> For a description of how this chip implements SWT reset requests resulting from SWT invalid accesses, see the chip-specific SWT information.<br><br>0 Invalid access to the SWT generates a bus error<br>1 Invalid access to the SWT causes a reset request if WEN is 1                                                                                       |
| 24<br>WND        | Window Mode<br>0 Regular mode: service sequence can be done at any time<br>1 Windowed mode: service sequence is valid only when the down counter is less than the value in the SWT_WN register                                                                                                                                                                                                            |
| 25<br>ITR        | Interrupt Then Reset Request<br><br><b>NOTE:</b> For a description of how this chip implements SWT reset requests and interrupt requests resulting from SWT time-outs, see the chip-specific SWT information.<br><br>0 Generate a reset request on a time-out<br>1 Generate an interrupt on an initial time-out; generate a reset request on a second consecutive time-out                                |
| 26<br>HLK        | Hard Lock<br>This bit is cleared only at reset.<br><br>0 SWT_CR, SWT_TO, SWT_WN and SWT_SK are read/write registers if SLK is 0<br>1 SWT_CR, SWT_TO, SWT_WN and SWT_SK are read-only registers                                                                                                                                                                                                            |
| 27<br>SLK        | Soft Lock<br>This bit is cleared by writing the unlock sequence to the service register.<br><br>0 SWT_CR, SWT_TO, SWT_WN and SWT_SK are read/write registers if HLK is 0<br>1 SWT_CR, SWT_TO, SWT_WN and SWT_SK are read-only registers                                                                                                                                                                   |
| 28<br>CSL        | Clock Selection<br>Selects the clock that drives the internal timer.<br><br>0 System clock<br>1 Oscillator clock                                                                                                                                                                                                                                                                                          |

Table continues on the next page...

## SWT\_CR field descriptions (continued)

| Field     | Description                                                                                                                                                                                    |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 29<br>STP | Stop Mode Control<br>Allows the watchdog timer to be stopped when the device enters stop mode.<br><br>0 SWT counter continues to run in stop mode<br>1 SWT counter is stopped in stop mode     |
| 30<br>FRZ | Debug Mode Control<br>Allows the watchdog timer to be stopped when the device enters debug mode.<br><br>0 SWT counter continues to run in debug mode<br>1 SWT counter is stopped in debug mode |
| 31<br>WEN | Watchdog Enabled<br><br>0 SWT is disabled<br>1 SWT is enabled                                                                                                                                  |

## 32.4.2 SWT Interrupt Register (SWT\_IR)

The SWT\_IR contains the time-out interrupt flag.

Address: 0h base + 4h offset = 4h

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |     |  |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--|--|--|--|--|--|--|-----|--|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |  |  |  |  |  |  |  |     |  |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |     |  |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |     |  |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |  |  |  |  |  |  |  |     |  |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |  |  |  |  |  |  |  |     |  |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  | TIF |  |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  | w1c |  |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |  |  |  |  |  |  |  |     |  |

## SWT\_IR field descriptions

| Field            | Description                                                                                                                                                                                          |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–30<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                              |
| 31<br>TIF        | Time-out Interrupt Flag<br>The flag and interrupt are cleared by writing a 1 to this bit. Writing a 0 has no effect.<br><br>0 No interrupt request<br>1 Interrupt request due to an initial time-out |

### 32.4.3 SWT Time-out Register (SWT\_TO)

#### NOTE

The reset value of this register is implementation specific. See the chip-specific SWT information.

The SWT Time-out (SWT\_TO) register contains the 32-bit time-out period. This register is read-only if either SWT\_CR[HLK] or SWT\_CR[SLK] is 1.

Address: 0h base + 8h offset = 8h

|       |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0   | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | WTO |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0*  | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* |    |

\* Notes:

- 

#### SWT\_TO field descriptions

| Field       | Description                                                                                                                                                                                          |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>WTO | Watchdog time-out period in clock cycles. An internal 32-bit down counter is loaded with this value or 0x100, whichever is greater, when the service sequence is written or when the SWT is enabled. |

### 32.4.4 SWT Window Register (SWT\_WN)

The SWT Window (SWT\_WN) register contains the 32-bit window start value. This register is cleared on reset. This register is read-only if either SWT\_CR[HLK] or SWT\_CR[SLK] is 1.

Address: 0h base + Ch offset = Ch

|       |     |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-----|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | WST |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |     |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

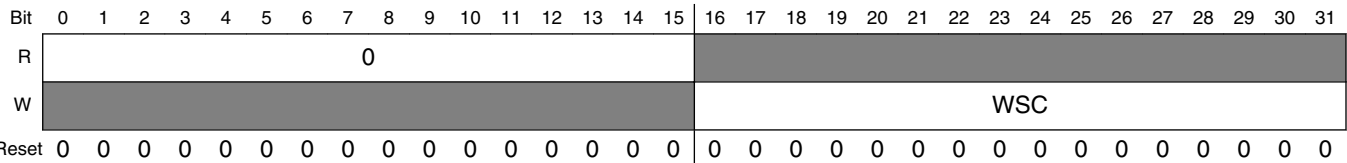
#### SWT\_WN field descriptions

| Field       | Description                                                                                                                                             |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>WST | Window Start Value<br><br>When window mode is enabled, the service sequence can only be written when the internal down counter is less than this value. |

32.4.5 SWT Service Register (SWT\_SR)

The SWT Service (SWT\_SR) register is the target for service operation writes used to reset the watchdog timer.

Address: 0h base + 10h offset = 10h



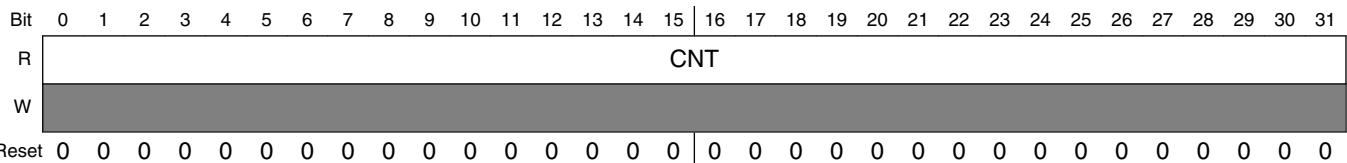
SWT\_SR field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 16–31<br>WSC     | Watchdog Service Code<br><br>This field is used to service the watchdog and to clear the Soft Lock bit (SWT_CR[SLK]). If the SWT_CR[SMD] field is 01b, two pseudorandom key values are written to service the watchdog; see <a href="#">Functional description</a> for details. Otherwise, the sequence 0xA602 followed by 0xB480 is written to the WSC field. To clear the Soft Lock bit (SWT_CR[SLK]), the value 0xC520 followed by 0xD928 is written to the WSC field. When read, the WSC field always returns zero. |

32.4.6 SWT Counter Output Register (SWT\_CO)

The SWT Counter Output (SWT\_CO) register is a read-only register that shows the value of the internal down counter when the SWT is disabled.

Address: 0h base + 14h offset = 14h



SWT\_CO field descriptions

| Field       | Description                                                                                                                                                                                                           |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>CNT | Watchdog Count<br><br>When the watchdog is disabled (SWT_CR[WEN] is 0), this field shows the value of the internal down counter. When the watchdog is enabled (SWT_CR[WEN] is 1), this field is cleared (the value is |

**SWT\_CO field descriptions (continued)**

| Field | Description                                                                                                                                                                                                                                                                               |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | 0x0000_0000). Values in this field can lag behind the internal counter value for up to 6 system clock cycles plus 8 counter clock cycles. Therefore, the value read from this field immediately after disabling the watchdog may be higher than the actual value of the internal counter. |

**32.4.7 SWT Service Key Register (SWT\_SK)**

The SWT Service Key (SWT\_SK) register holds the previous (or initial) service key value. This register is read-only if either SWT\_CR[HLK] or SWT\_CR[SLK] is 1.

Address: 0h base + 18h offset = 18h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | SK |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**SWT\_SK field descriptions**

| Field            | Description                                                                                                                                                                                                          |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                              |
| 16–31<br>SK      | Service Key<br><br>This field is the previous (or initial) service key value used in keyed service mode. If SWT_CR[SMD] is 01b, the next key value to be written to the SWT_SR is $(17 \cdot SK + 3) \bmod 2^{16}$ . |

**32.5 Functional description****32.5.1 Introduction**

The SWT is a 32-bit window watchdog timer designed to enable the system to recover in situations such as software getting trapped in a loop or if a bus transaction fails to terminate. It includes:

- a control register (SWT\_CR),
- an interrupt register (SWT\_IR),
- a time-out register (SWT\_TO),
- a window register (SWT\_WN),
- a service register (SWT\_SR),

- a counter output register (SWT\_CO), and
- a service key register (SWT\_SK).

Accesses to SWT registers occur with no peripheral bus wait states. (The peripheral bus bridge may add one or more system wait states.) However, due to synchronization logic in the SWT design, recognition of the service sequence or configuration changes may require up to three system plus seven counter clock cycles.

### **32.5.1.1 SWT\_CR register**

The SWT\_CR register includes bits to enable the timer, set configuration options, and lock configuration of the module. The watchdog is enabled by setting the SWT\_CR[WEN] bit. The reset value of the SWT\_CR[WEN] bit is device specific.<sup>3</sup> If the reset value of this bit is 1, the watchdog starts operation automatically after reset is released. Some devices can be configured to clear this bit automatically during the boot process.

### **32.5.1.2 SWT\_TO register**

The SWT\_TO register holds the watchdog time-out period in clock cycles unless the value is less than 0x100, in which case the time-out period is set to 0x100. When the SWT is enabled, the time-out period is loaded into an internal 32-bit down counter each time a valid service operation is performed. The SWT\_CR[CSL] bit selects which clock (system or oscillator) is used to drive the down counter.

### **32.5.1.3 SWT\_CO register**

The SWT\_CO register shows the value of the down counter when the watchdog is disabled. When the watchdog is enabled this register is cleared. The value shown in this register can lag behind the value in the internal counter for up to six system plus eight counter clock cycles.

The SWT\_CO register can be used during a software self test of the SWT. For example, the SWT can be enabled and not serviced for a fixed period of time less than the time-out value. Then the SWT can be disabled (SWT\_CR[WEN] cleared) and the value of the SWT\_CO register read to determine if the internal down counter is working properly.

---

3. See the chip-specific SWT information.

## 32.5.2 Configuration locking

The configuration of the SWT can be locked through use of either a soft lock or a hard lock. In either case, when locked, the SWT\_CR, SWT\_TO, SWT\_WN and SWT\_SK registers are read-only.

### 32.5.2.1 Hard lock

The hard lock is enabled by setting the SWT\_CR[HLK] bit, which can only be cleared by a reset.

### 32.5.2.2 Soft lock

The soft lock is enabled by setting the SWT\_CR[SLK] bit and is cleared by writing the unlock sequence to the service register.

## 32.5.3 Unlock sequence

The unlock sequence is a write of 0xC520 followed by a write of 0xD928 to the SWT\_SR[WSC] field. There is no timing requirement between the two writes. The unlock sequence logic ignores service sequence writes and recognizes the 0xC520, 0xD928 sequence regardless of previous writes. The unlock sequence can be written at any time and does not require the SWT\_CR[WEN] bit to be set.

## 32.5.4 Servicing operations

When enabled, the SWT requires periodic execution of a servicing operation that is determined by the SWT\_CR[SMD] field. Properly servicing the watchdog loads the internal down counter with the time-out period. The servicing modes are:

- fixed service sequence
- keyed service sequence

### 32.5.4.1 Fixed service sequence mode

If the SWT\_CR[SMD] field is 00b, the fixed service sequence mode is selected, which requires writing 0xA602, then 0xB480 to the SWT\_SR[WSC] field to service the watchdog. There is no timing requirement between the two writes and the service sequence logic ignores unlock sequence writes.

### 32.5.4.2 Keyed service sequence mode

If the SWT\_CR[SMD] field is 01b, then the keyed service sequence mode is selected, which requires writing two pseudorandom keys to the SWT\_SR[WSC] field to service the watchdog. The key values are determined by the pseudorandom key generator defined by the equation in the following figure. This algorithm will generate a sequence of  $2^{16}$  different key values before repeating. The state of the key generator is held in the SWT\_SK register. For example, if SWT\_SK[SK] is 0x0100, then the service sequence keys are 0x1103, 0x2136. In this mode, each time a valid key is written to the SWT\_SR register, the SWT\_SK register is updated. So, after servicing the watchdog by writing 0x1103 and then 0x2136 to the SWT\_SR[WSC] field, SWT\_SK[SK] is 0x2136 and the next key sequence is 0x3499, 0x7E2C.

$$SK_{n+1} = (17 \times SK_n + 3) \bmod 2^{16}$$

**Figure 32-8. Pseudorandom Key Generator**

### 32.5.4.3 Window mode

If window mode is enabled (SWT\_CR[WND] bit is 1), the service sequence must be applied in the last part of the time-out period defined by SWT\_WN.

- The window is open when the down counter is less than the value in SWT\_WN.
- Outside this window, service sequence writes are invalid accesses and generate a bus error or reset request depending on the value of SWT\_CR[RIA].

For example, if SWT\_TO is set to 5000 and SWT\_WN is set to 1000, the service sequence must be applied in the last 20% of the time-out period.

Synchronization logic in the watchdog design slightly delays when the window opens. This delay could be up to 3 system clock cycles plus 4 counter clock cycles.



### 32.5.5 Time-out

The SWT\_CR[ITR] bit controls the SWT's action taken when a time-out occurs.

- If the SWT\_CR[ITR] bit is 0, the SWT generates a reset request immediately on any time-out.
- If the SWT\_CR[ITR] bit is 1, an initial time-out causes the SWT to generate an interrupt and load the down counter with the time-out period.
  - The interrupt is indicated by the Time-out Interrupt Flag (SWT\_IR[TIF]).
  - Clear the interrupt by writing 1 to SWT\_IR[TIF].

If the service sequence is not written before the second consecutive time-out, the SWT generates a reset request.

### 32.5.6 Initialization

All registers should be initialized before setting the SWT\_CR[WEN] bit to enable the watchdog. Registers can be initialized in any sequence.



## Chapter 33

# Periodic Interrupt Timer (PIT-RTI)

### 33.1 Chip-specific PIT information

This chip has one instance of the PIT.

PIT channels can be used to trigger periodic ADC conversions via EQADC Command FIFOs. For more information, see:

- [eQADC Command FIFO Trigger Source Registers \(SIU\\_ISEL4\) to eQADC Command FIFO Trigger Source Registers \(SIU\\_ISEL7\) and eQADC Advance Trigger Source Register \(SIU\\_ISEL9\)](#)
- [SIU Trigger Burst Generators \(TBG\)](#)

### 33.2 Introduction

The PIT module is an array of timers that can be used to raise interrupts and triggers. It includes a dedicated Real Time Interrupt Timer (RTI), which runs on a separate clock and can be used for system wakeup

#### 33.2.1 Block diagram

The following figure shows the block diagram of the PIT-RTI module.

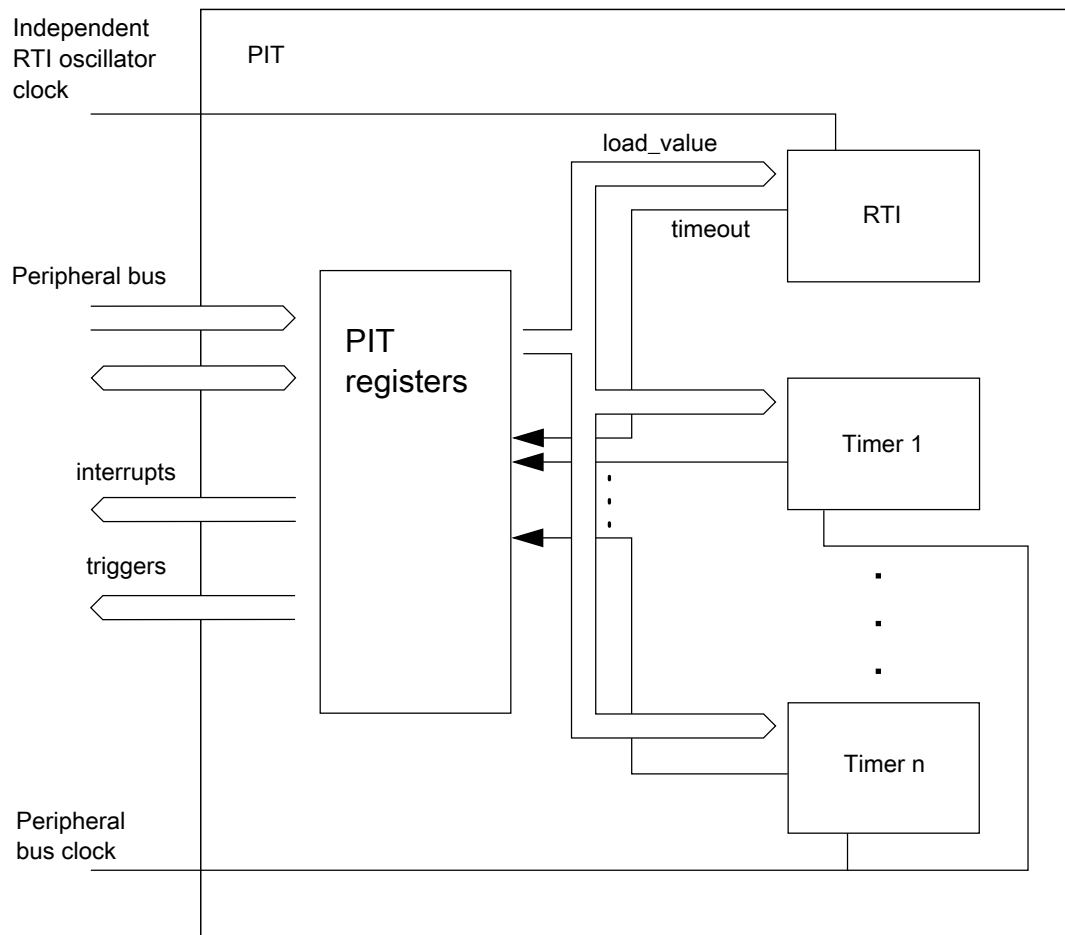


**Figure 33-1. Block diagram of the PIT-RTI**

### NOTE

See the chip-specific PIT information for the number of PIT channels used in this MCU.

## 33.2.2 Features

The main features of this block are:

- One RTI (Real-Time Interrupt) timer to wakeup the CPU
- Ability of timers to generate trigger pulses
- Ability of timers to generate interrupts
- Maskable interrupts
- Option to raise RTI interrupt, even when the bus clock is switched off

- Power saving with a separate input clock for the RTI timer. All other timers share one common core clock.
- Independent timeout periods for each timer

### 33.3 Signal description

The PIT module has no external pins.

### 33.4 Memory map/register description

This section provides a detailed description of all registers accessible in the PIT\_RTIC module.

- See the chip-specific PIT information for the number of PIT channels used in this MCU.
- The RTI registers should be programmed only when the RTI clock is running.

#### NOTE

Write request to a register must have the byte enable asserted for the bytes user wants to update.

#### PIT memory map

| Address offset (hex) | Register name                                   | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|-------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 0                    | PIT Module Control Register (PIT_MCR)           | 32              | R/W    | 0000_0000h  | <a href="#">33.4.1/1178</a>  |
| E0                   | PIT Upper Lifetime Timer Register (PIT_LTMR64H) | 32              | R      | 0000_0000h  | <a href="#">33.4.2/1179</a>  |
| E4                   | PIT Lower Lifetime Timer Register (PIT_LTMR64L) | 32              | R      | 0000_0000h  | <a href="#">33.4.3/1180</a>  |
| F0                   | Timer Load Value Register (PIT_RTIL_DVAL)       | 32              | R/W    | 0000_0000h  | <a href="#">33.4.4/1180</a>  |
| F4                   | Current Timer Value Register (PIT_RTIL_CVAL)    | 32              | R      | 0000_0000h  | <a href="#">33.4.5/1181</a>  |
| F8                   | Timer Control Register (PIT_RTIL_TCTRL)         | 32              | R/W    | 0000_0000h  | <a href="#">33.4.6/1181</a>  |
| FC                   | Timer Flag Register (PIT_RTIL_TFLG)             | 32              | R/W    | 0000_0000h  | <a href="#">33.4.7/1182</a>  |
| 100                  | Timer Load Value Register (PIT_LDVAL0)          | 32              | R/W    | 0000_0000h  | <a href="#">33.4.8/1183</a>  |
| 104                  | Current Timer Value Register (PIT_CVAL0)        | 32              | R      | 0000_0000h  | <a href="#">33.4.9/1183</a>  |
| 108                  | Timer Control Register (PIT_TCTRL0)             | 32              | R/W    | 0000_0000h  | <a href="#">33.4.10/1184</a> |
| 10C                  | Timer Flag Register (PIT_TFLG0)                 | 32              | R/W    | 0000_0000h  | <a href="#">33.4.11/1185</a> |
| 110                  | Timer Load Value Register (PIT_LDVAL1)          | 32              | R/W    | 0000_0000h  | <a href="#">33.4.8/1183</a>  |
| 114                  | Current Timer Value Register (PIT_CVAL1)        | 32              | R      | 0000_0000h  | <a href="#">33.4.9/1183</a>  |

Table continues on the next page...

## PIT memory map (continued)

| Address offset (hex) | Register name                            | Width (in bits) | Access | Reset value | Section/page                 |
|----------------------|------------------------------------------|-----------------|--------|-------------|------------------------------|
| 118                  | Timer Control Register (PIT_TCTRL1)      | 32              | R/W    | 0000_0000h  | <a href="#">33.4.10/1184</a> |
| 11C                  | Timer Flag Register (PIT_TFLG1)          | 32              | R/W    | 0000_0000h  | <a href="#">33.4.11/1185</a> |
| 120                  | Timer Load Value Register (PIT_LDVAL2)   | 32              | R/W    | 0000_0000h  | <a href="#">33.4.8/1183</a>  |
| 124                  | Current Timer Value Register (PIT_CVAL2) | 32              | R      | 0000_0000h  | <a href="#">33.4.9/1183</a>  |
| 128                  | Timer Control Register (PIT_TCTRL2)      | 32              | R/W    | 0000_0000h  | <a href="#">33.4.10/1184</a> |
| 12C                  | Timer Flag Register (PIT_TFLG2)          | 32              | R/W    | 0000_0000h  | <a href="#">33.4.11/1185</a> |
| 130                  | Timer Load Value Register (PIT_LDVAL3)   | 32              | R/W    | 0000_0000h  | <a href="#">33.4.8/1183</a>  |
| 134                  | Current Timer Value Register (PIT_CVAL3) | 32              | R      | 0000_0000h  | <a href="#">33.4.9/1183</a>  |
| 138                  | Timer Control Register (PIT_TCTRL3)      | 32              | R/W    | 0000_0000h  | <a href="#">33.4.10/1184</a> |
| 13C                  | Timer Flag Register (PIT_TFLG3)          | 32              | R/W    | 0000_0000h  | <a href="#">33.4.11/1185</a> |

## 33.4.1 PIT Module Control Register (PIT\_MCR)

This register enables or disables the PIT timer clocks and controls the timers when the PIT enters the Debug mode.

Access: User read/write

Address: 0h base + 0h offset = 0h

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |          |      |     |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--|--|--|--|--|--|--|--|--|--|--|--|----------|------|-----|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |  |  |  |  |  |  |  |  |  |  |  |  |          |      |     |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |          |      |     |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |          |      |     |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |  |  |  |  |  |  |  |  |  |  |  |  |          |      |     |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |  |  |  |  |  |  |  |  |  |  |  |  |          |      |     |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  | MDIS_RTI | MDIS | FRZ |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |          |      |     |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |  |  |  |  |  |  |  |  |  |  |  |  |          |      |     |

**PIT\_MCR field descriptions**

| Field            | Description                                                                                                                                                                                                                                                                    |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–28<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                        |
| 29<br>MDIS_RTI   | Module Disable - RTI section<br><br>This is used to disable the RTI timer. This bit must be enabled before any RTI setup is done.<br><br>0 Clock for RTI Timers is enabled<br>1 Clock for RTI Timers is disabled                                                               |
| 30<br>MDIS       | Module Disable - (PIT section)<br><br>Disables the standard timers. The RTI timer is not affected by this field. This field must be enabled before any other setup is done.<br><br>0 Clock for standard PIT timers is enabled.<br>1 Clock for standard PIT timers is disabled. |
| 31<br>FRZ        | Freeze<br><br>Allows the timers to be stopped when the device enters the Debug mode.<br><br>0 Timers continue to run in Debug mode.<br>1 Timers are stopped in Debug mode.                                                                                                     |

**33.4.2 PIT Upper Lifetime Timer Register (PIT\_LTMR64H)**

This register is intended for applications that chain timer 0 and timer 1 to build a 64-bit lifetimer.

Access: User read only

Address: 0h base + E0h offset = E0h

|       |     |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-----|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | LTH |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |     |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**PIT\_LTMR64H field descriptions**

| Field       | Description                                                                                                                                     |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>LTH | Life Timer value<br><br>Shows the timer value of timer 1. If this register is read at a time t1, LTMR64L shows the value of timer 0 at time t1. |

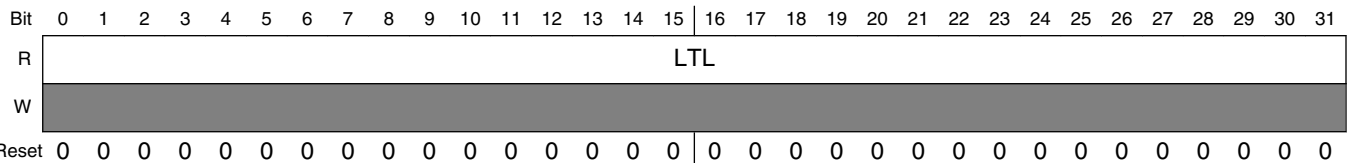
33.4.3 PIT Lower Lifetime Timer Register (PIT\_LTMR64L)

This register is intended for applications that chain timer 0 and timer 1 to build a 64-bit lifetimer.

To use LTMR64H and LTMR64L, timer 0 and timer 1 need to be chained. To obtain the correct value, first read LTMR64H and then LTMR64L. LTMR64H will have the value of CVAL1 at the time of the first access, LTMR64L will have the value of CVAL0 at the time of the first access, therefore the application does not need to worry about carry-over effects of the running counter.

Access: User read only

Address: 0h base + E4h offset = E4h



PIT\_LTMR64L field descriptions

| Field       | Description                                                                                                                   |
|-------------|-------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>LTL | Life Timer value<br><br>Shows the value of timer 0 at the time LTMR64H was last read. It will only update if LTMR64H is read. |

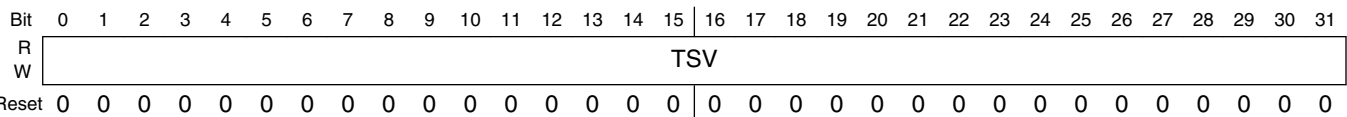
33.4.4 Timer Load Value Register (PIT\_RTI\_LDVAL)

These registers select the timeout period for the timer interrupts.

In the case of the RTI, it will take several cycles until this value is synchronized into the RTI clock domain. For all other timers the value change is visible immediately. The synchronisation mechanism allows 0 wait states in this case.

Access: User read/write

Address: 0h base + F0h offset = F0h





**PIT\_RTI\_LDVAL field descriptions**

| Field       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>TSV | <p>Timer Start Value</p> <p>Sets the timer start value. The timer will count down until it reaches 0, then it will generate an interrupt and load this register value again. Writing a new value to this register will not restart the timer; instead the value will be loaded after the timer expires. To abort the current cycle and start a timer period with the new value, the timer must be disabled and enabled again.</p> <p><b>NOTE:</b> The RTI timer must not be set to a value lower than 32 cycles, otherwise interrupts may be lost, as it takes several cycles to clear the RTI interrupt. For the other timers, this limit does not apply, however there will be practical limits, since the processor will require several cycles to service an interrupt.</p> |

**33.4.5 Current Timer Value Register (PIT\_RTI\_CVAL)**

These registers indicate the current timer position.

For the RTI timers, they show a value which is several cycles old, since it originates from a potentially different clock domain.

Access: User read only

Address: 0h base + F4h offset = F4h

|       |     |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-----|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | TVL |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |     |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**PIT\_RTI\_CVAL field descriptions**

| Field       | Description                                                                                                                                                                                                                                                                                                                                           |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>TVL | <p>Current Timer Value</p> <p>Represents the current timer value, if the timer is enabled.</p> <p><b>NOTE:</b></p> <ul style="list-style-type: none"> <li>If the timer is disabled, do not use this field as its value is unreliable.</li> <li>The timer uses a downcounter. The timer values are frozen in Debug mode if MCR[FRZ] is set.</li> </ul> |

**33.4.6 Timer Control Register (PIT\_RTI\_TCTRL)**

These registers contain the control bits for each timer.

Access: User read/write

## Memory map/register description

Address: 0h base + F8h offset = F8h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |     |     |     |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|-----|-----|
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29  | 30  | 31  |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |    | CHN | TIE | TEN |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |     |     |     |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0   | 0   |

### PIT\_RTI\_TCTRL field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                   |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–28<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                       |
| 29<br>CHN        | Chain Mode<br>The RTI timer cannot be chained, writing to this bit has no impact.                                                                                                                                                                                                                                                             |
| 30<br>TIE        | Timer Interrupt Enable<br>When an interrupt is pending or, RTI_TFLG[TIF] is set, enabling the interrupt will immediately cause an interrupt event. To avoid this, the associated RTI_TFLG[TIF] flag must be cleared first.<br><br>0 Interrupt requests from the RTI Timer are disabled.<br>1 Interrupt will be requested whenever TIF is set. |
| 31<br>TEN        | Timer Enable Bit<br>This bit enables or disables the timer.<br><br>0 RTI Timer is disabled.<br>1 RTI Timer is active.                                                                                                                                                                                                                         |

## 33.4.7 Timer Flag Register (PIT\_RTI\_TFLG)

These registers hold the PIT interrupt flags.

Access: User read/write

Address: 0h base + FCh offset = FCh

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31  |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    | TIF |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | w1c |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   |

**PIT\_RTI\_TFLG field descriptions**

| Field            | Description                                                                                                                                                                                                                                                                          |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–30<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                              |
| 31<br>TIF        | Timer Interrupt Flag.<br><br>TIF is set to 1 at the end of the timer period. This flag can be cleared only by writing it with 1. Writing 0 has no effect. If enabled (TIE = 1), TIF causes an interrupt request.<br><br>0 Time-out has not yet occurred.<br>1 Time-out has occurred. |

**33.4.8 Timer Load Value Register (PIT\_LDVALn)**

These registers select the timeout period for the timer interrupts.

Access: User read/write

Address: 0h base + 100h offset + (16d × i), where i=0d to 3d

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**PIT\_LDVALn field descriptions**

| Field       | Description                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>TSV | Timer Start Value<br><br>Sets the timer start value. The timer will count down until it reaches 0, then it will generate an interrupt and load this register value again. Writing a new value to this register will not restart the timer; instead the value will be loaded after the timer expires. To abort the current cycle and start a timer period with the new value, the timer must be disabled and enabled again. |

**33.4.9 Current Timer Value Register (PIT\_CVALn)**

These registers indicate the current timer position.

Access: User read only

Address: 0h base + 104h offset + (16d × i), where i=0d to 3d

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

PIT\_CVAL<sub>n</sub> field descriptions

| Field       | Description                                                                                                                                                                                                                                                                                                                                           |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>TVL | <p>Current Timer Value</p> <p>Represents the current timer value, if the timer is enabled.</p> <p><b>NOTE:</b></p> <ul style="list-style-type: none"> <li>If the timer is disabled, do not use this field as its value is unreliable.</li> <li>The timer uses a downcounter. The timer values are frozen in Debug mode if MCR[FRZ] is set.</li> </ul> |

33.4.10 Timer Control Register (PIT\_TCTRL<sub>n</sub>)

These registers contain the control bits for each timer.

Access: User read/write

Address: 0h base + 108h offset + (16d × i), where i=0d to 3d

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |     |  |     |  |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----|-----|--|-----|--|
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29  | 30 | 31  |  |     |  |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |    | CHN |    | TIE |  | TEN |  |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |     |  |     |  |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0  | 0   |  |     |  |

PIT\_TCTRL<sub>n</sub> field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                  |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–28<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                           |
| 29<br>CHN        | <p>Chain Mode</p> <p>When activated, Timer n-1 needs to expire before timer n can decrement by 1.</p> <p>Timer 0 cannot be chained. The RTI timer cannot be chained.</p> <p>0 Timer is not chained.</p> <p>1 Timer is chained to previous timer. For example, for Channel 2, if this field is set, Timer 2 is chained to Timer 1.</p>        |
| 30<br>TIE        | <p>Timer Interrupt Enable</p> <p>When an interrupt is pending, or, TFLGn[TIF] is set, enabling the interrupt will immediately cause an interrupt event. To avoid this, the associated TFLGn[TIF] must be cleared first.</p> <p>0 Interrupt requests from Timer n are disabled.</p> <p>1 Interrupt will be requested whenever TIF is set.</p> |
| 31<br>TEN        | <p>Timer Enable</p> <p>Enables or disables the timer.</p>                                                                                                                                                                                                                                                                                    |

Table continues on the next page...

**PIT\_TCTRLn field descriptions (continued)**

| Field | Description          |
|-------|----------------------|
| 0     | Timer n is disabled. |
| 1     | Timer n is enabled.  |

**33.4.11 Timer Flag Register (PIT\_TFLGn)**

These registers hold the PIT interrupt flags.

Access: User read/write

Address: 0h base + 10Ch offset + (16d × i), where i=0d to 3d

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |     |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--|--|--|--|--|--|--|--|--|--|--|--|--|--|-----|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |     |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |     |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |     |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |     |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |     |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  | TIF |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  | w1c |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |     |

**PIT\_TFLGn field descriptions**

| Field            | Description                                                                                                                                                                                                                                                               |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–30<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                   |
| 31<br>TIF        | Timer Interrupt Flag<br><br>Sets to 1 at the end of the timer period. Writing 1 to this flag clears it. Writing 0 has no effect. If enabled, or, when TCTRLn[TIE] = 1, TIF causes an interrupt request.<br><br>0 Timeout has not yet occurred.<br>1 Timeout has occurred. |

**33.5 Functional description**

This section provides the functional description of the module.

### 33.5.1 General operation

This section gives detailed information on the internal operation of the module. Each timer can be used to generate trigger pulses and interrupts. Each interrupt is available on a separate interrupt line. Additionally the RTI timer can be used to wake up the processor.

#### 33.5.1.1 Timers / RTI

The timers generate triggers at periodic intervals, when enabled. The timers load the start values as specified in their LDVAL registers, count down to 0 and then load the respective start value again. Each time a timer reaches 0, it will generate a trigger pulse and set the interrupt flag.

All interrupts can be enabled or masked by setting TCTRLn[TIE]. A new interrupt can be generated only after the previous one is cleared. Since in the case of the RTI, clearing the interrupt crosses clock domains, a minimum load value of 32 must be maintained.

If desired, the current counter value of the timer can be read via the CVAL registers. The value of the RTI counter can be delayed considerably, as it is synchronized to the bus clock from the RTI clock domain.

The counter period can be restarted, by first disabling, and then enabling the timer with TCTRLn[TEN]. See the following figure.

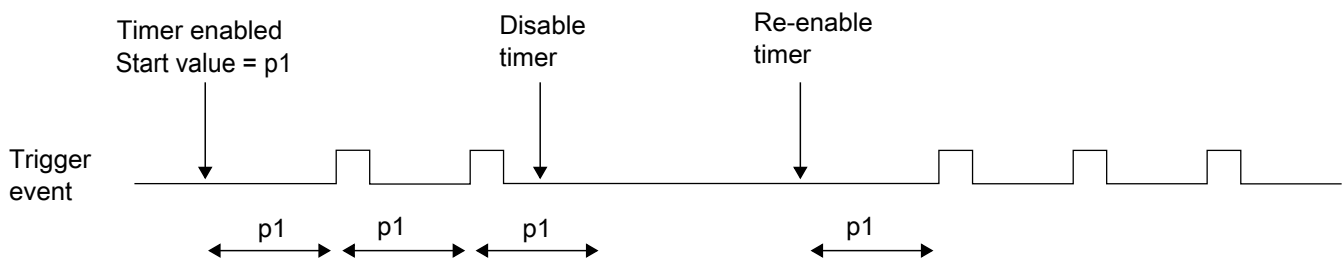


**Figure 33-29. Stopping and starting a timer**

The counter period of a running timer can be modified, by first disabling the timer, setting a new load value, and then enabling the timer again. See the following figure.

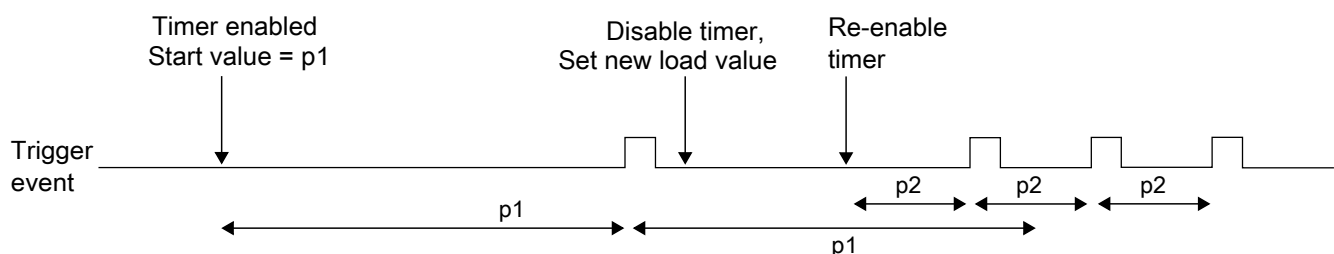


**Figure 33-30. Modifying running timer period**

It is also possible to change the counter period without restarting the timer by writing LDVAL with the new load value. This value will then be loaded after the next trigger event. See the following figure.

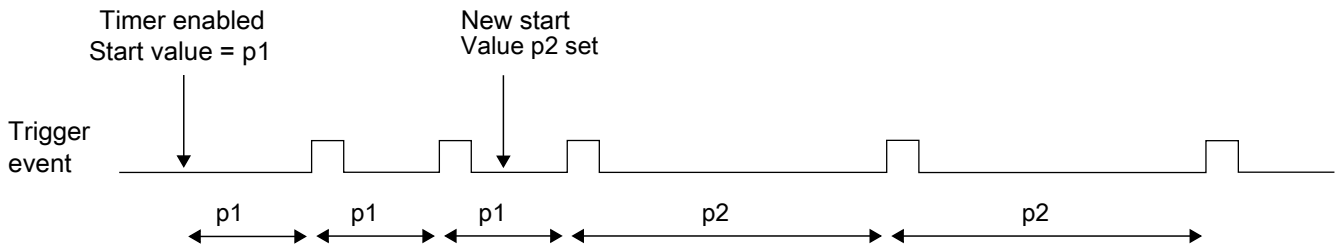


**Figure 33-31. Dynamically setting a new load value**

### NOTE

- If the pause is initiated, when the timer is nearing 0 (CVALn = 0x0), the pause command may not make it to the IP before the timer expires and generates a trigger.
- Pause will be ignored if (CVALn = 0x0), but the trigger will remain asserted until the pause is removed. The user is recommended to remove Pause and then clear the interrupt TFLGn[TIF].
- If the timers are to be paused, sufficient time must be ensured for the IP to react.

### 33.5.1.2 Debug mode

In Debug mode, the timers will be frozen based on MCR[FRZ]. This is intended to aid software development, allowing the developer to halt the processor, investigate the current state of the system, for example, the timer values, and then continue the operation.

## 33.5.2 Interrupts

All the timers support interrupt generation. The RTI is typically used for system wakeup, but can be used for interrupt generation as well. See the MCU specification for related vector addresses and priorities.

Timer interrupts can be enabled by setting TCTRLn[TIE]. TFLGn[TIF] are set to 1 when a timeout occurs on the associated timer, and are cleared to 0 by writing a 1 to the corresponding TFLGn[TIF].

The PIT\_RTI generates a real time interrupt when the selected interrupt time period elapses. The RTI interrupt is disabled locally by setting TCTCRLn[TIE] to zero. The real time interrupt flag (TIF) is set to 1 when a timeout occurs, and is cleared by writing a 1 to the corresponding TFLGn[TIF]. The flag will be set regardless whether the interrupt is enabled.

The RTI can be used for periodic wakeup from a low-power mode. It can also be used to generate a general-purpose interrupt.

### 33.5.3 Chained timers

When a timer has chain mode enabled, it will only count after the previous timer has expired. So if timer n-1 has counted down to 0, counter n will decrement the value by one. This allows to chain some of the timers together to form a longer timer. The first timer (timer 0) cannot be chained to any other timer.

## 33.6 Initialization and application information

In the example configuration:

- The PIT clock has a frequency of 50 MHz.
- The RTI clock has a frequency of 10 MHz.
- The RTI creates a wakeup interrupt every 500 ms.
- Timer 1 creates an interrupt every 5.12 ms.
- Timer 3 creates a trigger event every 30 ms.

The PIT module must be activated by writing a 0 to MCR[MDIS].

The 50 MHz clock frequency equates to a clock period of 20 ns and the 10 MHz frequency equates to a clock period of 100 ns. Therefore, the RTI timer needs to trigger every  $500 \text{ ms} / 100 \text{ ns} = 5,000,000$  cycles. Timer 1 needs to trigger every  $5.12 \text{ ms} / 20 \text{ ns} = 256,000$  cycles and Timer 3 every  $30 \text{ ms} / 20 \text{ ns} = 1,500,000$  cycles. The value for the LDVAL register trigger is calculated as:

$\text{LDVAL trigger} = (\text{period} / \text{clock period}) - 1$

This means RTI LDVAL will be written with 004C4B3F hex; LDVAL1 and LDVAL3 must be written with 0x0003E7FF and 0x0016E35F respectively.



To generate the wakeup interrupt, the interrupt line must be enabled by writing 1 to the RTI TIE bit in the TCTRL register. To start the RTI, the TEN in the RTI TCTRL register must also be set.

The interrupt for Timer 1 is enabled by setting TCTRL1[TIE]. The timer is started by writing 1 to TCTRL1[TEN].

Timer 3 shall be used only for triggering. Therefore, Timer 3 is started by writing a 1 to TCTRL3[TEN]. TCTRL3[TIE] stays at 0.

The following example code matches the described setup:

```
// turn on PIT
PIT_MCR = 0x00;

// RTI
PIT_RTI_LDVAL = 0x004C4B3F; // setup RTI for 5000000 cycles
PIT_RTI_TCTRL = PIT_TIE; // let RTI generate interrupts
PIT_RTI_TCTRL |= PIT_TEN; // start RTI

// Timer 1
PIT_LDVAL1 = 0x0003E7FF; // setup timer 1 for 256000 cycles
PIT_TCTRL1 = TIE; // enable Timer 1 interrupts
PIT_TCTRL1 |= TEN; // start Timer 1

// Timer 3
PIT_LDVAL3 = 0x0016E35F; // setup timer 3 for 1500000 cycles
PIT_TCTRL3 |= TEN; // start Timer 3
```

## 33.7 Example configuration for chained timers

In the example configuration:

- The PIT clock has a frequency of 100 MHz.
- Timers 1 and 2 are available.
- An interrupt shall be raised every 1 minute.

The PIT module needs to be activated by writing a 0 to MCR[MDIS].

The 100 MHz clock frequency equates to a clock period of 10 ns, so the PIT needs to count for 6000 million cycles, which is more than a single timer can do. So, Timer 1 is set up to trigger every 6 s (600 million cycles). Timer 2 is chained to Timer 1 and programmed to trigger 10 times.

The value for the LDVAL register trigger is calculated as number of cycles-1, so LDVAL1 receives the value 0x23C345FF and LDVAL2 receives the value 0x00000009.

The interrupt for Timer 2 is enabled by setting TCTRL2[TIE], the Chain mode is activated by setting TCTRL2[CHN], and the timer is started by writing a 1 to TCTRL2[TEN]. TCTRL1[TEN] needs to be set, and TCTRL1[CHN] and TCTRL1[TIE] are cleared.

The following example code matches the described setup:

```
// turn on PIT
PIT_MCR = 0x00;

// Timer 2
PIT_LDVAL2 = 0x00000009; // setup Timer 2 for 10 counts
PIT_TCTRL2 = TIE; // enable Timer 2 interrupt
PIT_TCTRL2 |= CHN; // chain Timer 2 to Timer 1
PIT_TCTRL2 |= TEN; // start Timer 2

// Timer 1
PIT_LDVAL1 = 0x23C345FF; // setup Timer 1 for 600 000 000 cycles
PIT_TCTRL1 = TEN; // start Timer 1
```

## 33.8 Example configuration for the lifetime timer

To configure the lifetime timer, channels 0 and 1 need to be chained together.

First the PIT module needs to be activated by writing a 0 to the MDIS bit in the CTRL register, then the LDVAL registers need to be set to the maximum value.

The timer is a downcounter.

The following example code matches the described setup:

```
// turn on PIT
PIT_MCR = 0x00;

// Timer 1
PIT_LDVAL1 = 0xFFFFFFFF; // setup timer 1 for maximum counting period
PIT_TCTRL1 = 0x0; // disable timer 1 interrupts
PIT_TCTRL1 |= CHN; // chain timer 1 to timer 0
PIT_TCTRL1 |= TEN; // start timer 1

// Timer 0
PIT_LDVAL0 = 0xFFFFFFFF; // setup timer 0 for maximum counting period
PIT_TCTRL0 = TEN; // start timer 0
```

To access the lifetime, read first LTMR64H and then LTMR64L.

```
current_uptime = PIT_LTMR64H<<32;
current_uptime = current_uptime + PIT_LTMR64L;
```

# Chapter 34

## Enhanced Modular IO Subsystem (eMIOS)

### 34.1 Chip-specific eMIOS information

This has two eMIOS module instances: eMIOS\_0 and eMIOS\_1. The two modules are identical and located on different Peripheral Bridges.

#### 34.1.1 eMIOS channels and counter buses

Each eMIOS module instance has 16 channels, for a total of 32 channels. The 32 channels are split across the two instances to support redundancy for safety applications.

The following table shows the mapping of the 32 eMIOS external balls to the instance channels.

**Table 34-1. eMIOS channel mapping**

| eMIOS balls | Instance channels |         |
|-------------|-------------------|---------|
|             | eMIOS_0           | eMIOS_1 |
| 0-7         | 0-7               | —       |
| 8-15        | —                 | 0-7     |
| 16-23       | 16-23             | —       |
| 24-31       | —                 | 16-23   |

For both module instances, each set of implemented channels drives a counter bus:

- Channels 0-7 drive counter bus B.
- Channels 16-23 drive counter bus D.

## 34.1.2 eMIOS unimplemented channels

For both module instances, channels 8-15 and 24-31 are not implemented. References to features and functionality related to these channels—including registers, register fields, and counter buses—do not apply to this chip.

## 34.1.3 eMIOS interaction with other modules

The eMIOS modules interface with eTPU, DSPI, and IGF modules.

### 34.1.3.1 eMIOS global time base synchronization with eTPU

The Global Time Base Enable (GTBE) signal output of the eMIOS module instances, which is controlled by the eMIOS's MCR[GTBE] field, is ORed together with the GTBE signal output from the two eTPU instances. The ORed signal is then fed back to the GTBE input of all eMIOS and eTPU instances.

These connections enable global time base synchronization through the following procedure:

1. Negate the GTBE field in all eMIOS and eTPU instances.
2. Configure eMIOS and eTPU channels.
3. Assert the GTBE field in one of the eMIOS or eTPU instances.

### 34.1.3.2 eMIOS and eTPU ODIS connections

The following table lists the Output Disable (ODIS) connections between eMIOS and eTPU modules. The eMIOS modules' Cn[ODISSL] fields for the applicable channels use these connections.

**Table 34-2. eMIOS and eTPU ODIS connections**

| Source   |                       | Destination |                 | Description                                                                                                   |
|----------|-----------------------|-------------|-----------------|---------------------------------------------------------------------------------------------------------------|
| Instance | Pin                   | Instance    | Pin             |                                                                                                               |
| EMIOS_0  | emios_flag_out[16:19] | EMIOS_0     | emios_odis[3:0] | Connection from EMIOS_0 Channel Flag Out numbers 16, 17, 18, 19 to EMIOS_0 ODIS bits 3, 2, 1, 0, respectively |
| EMIOS_1  | emios_flag_out[16:19] | EMIOS_1     | emios_odis[3:0] | Connection from EMIOS_1 Channel Flag Out numbers 16, 17, 18, 19 to EMIOS_1 ODIS bits 3, 2, 1, 0, respectively |

*Table continues on the next page...*

**Table 34-2. eMIOS and eTPU ODIS connections (continued)**

| Source   |                       | Destination |                  | Description                                                                                                  |
|----------|-----------------------|-------------|------------------|--------------------------------------------------------------------------------------------------------------|
| Instance | Pin                   | Instance    | Pin              |                                                                                                              |
| EMIOS_0  | emios_flag_out[16:19] | ETPU_C      | etpu_odis_1[3:0] | Connection from EMIOS_0 Channel Flag Out numbers 16, 17, 18, 19 to ETPU_C ODIS bits 3, 2, 1, 0, respectively |
| EMIOS_1  | emios_flag_out[16:19] | ETPU_A      | etpu_odis_1[3:0] | Connection from EMIOS_1 Channel Flag Out numbers 16, 17, 18, 19 to ETPU_A ODIS bits 3, 2, 1, 0, respectively |
| EMIOS_1  | emios_flag_out[23:20] | ETPU_B      | etpu_odis_2[3:0] | Connection from EMIOS_1 Channel Flag Out numbers 23, 22, 21, 20 to ETPU_B ODIS bits 3, 2, 1, 0, respectively |

### 34.1.3.3 STAC server IDs

STAC server IDs are defined in [STAC server IDs](#).

### 34.1.3.4 eMIOS and DSPI connections

See [DSPI and eMIOS direct connections](#) and [DSPI connections with SIU, eTPU, and eMIOS modules](#).

### 34.1.3.5 eMIOS inputs with Glitch Filters

See [IGF interaction with other modules](#) for connections between eMIOS and Input Glitch Filters (IGFs).

### 34.1.3.6 eMIOS and eTPU signal passthrough from pins via IGFs

On this chip, the IGF modules mediate between some pins and the eMIOS and eTPU modules. After system reset, the IGF modules are in Configuration mode by default. This mode disables IGF output and prevents the signals from the pins from reaching the eMIOS and eTPU modules.

The signals affected by this situation appear in [IGF interaction with other modules](#).

To route an affected signal directly to an eMIOS or eTPU instance, program the applicable channel of the applicable IGF instance:

1. Ensure that IGF\_MCRn[MDIS] is 0 and that the IGF channel is disabled: IGF\_MCRn[FGEN] is 0.
2. Configure the rising and falling edges for bypass:
  - Rising edge: Write 00b to IGF\_MCRn[RFM].
  - Falling edge: Write 00b to IGF\_MCRn[FFM].
3. Enable the IGF channel: Write 1 to IGF\_MCRn[FGEN].

At the end of this sequence, the programmed filter enters Normal mode and passes through the affected signal.

## **34.2 Introduction**

### **34.2.1 Overview**

The configurable enhanced Modular Input/Output Subsystem (eMIOS) provides functionality to generate or measure time events. Its overall architecture resembles that of the MIOS timer.

The predecessor MIOS timer block provides a framework in which a set of sublocks with different timer functions are assembled to attend to the specific needs of an SoC. The eMIOS builds on this concept by using a Unified Channel module that provides a superset of the functionality of all the individual MIOS channels, while also providing a consistent user interface. This allows more flexibility as each Unified Channel can be programmed for different functions in different applications of the SoC. In addition, eMIOS architecture allows the use of Dedicated Channels which perform specific functions not included in MIOS inheritance.

The block diagram of the configurable eMIOS block is shown below.

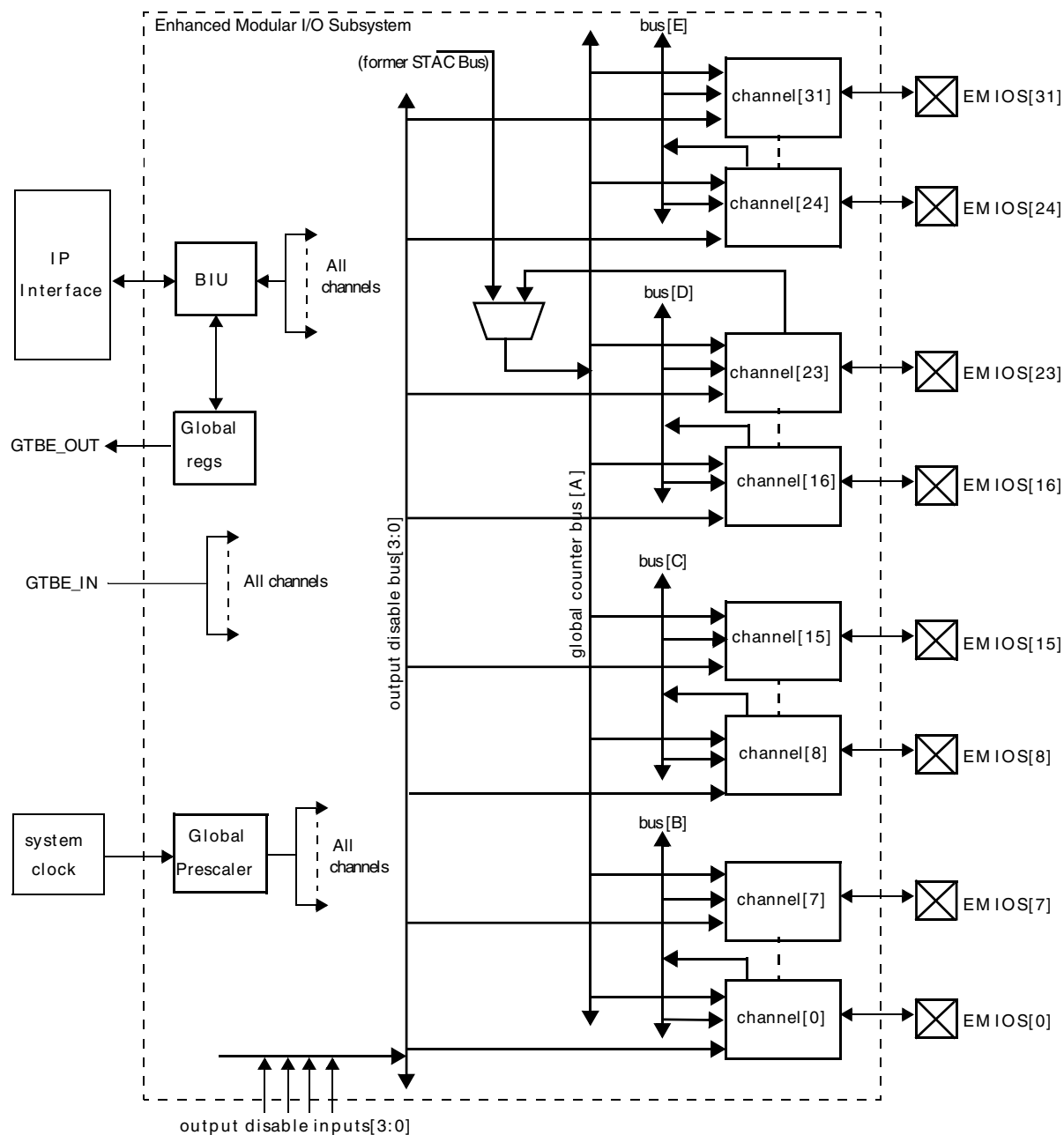


**Figure 34-1. eMIOS full channel configuration using Unified Channels only (see Configuration section)**

### 34.2.2 Features

The eMIOS has the following features:

- Up to 32 channels chosen among Unified or Dedicated Channels, not necessarily numbered in a continuous sequence. (See Configuration information)
- Data registers of either 8-, 16-, 24-, or 32-bit width. (See Configuration information)
- Counter buses B, C, D, and E can be driven by Unified Channels 0, 8, 16, and 24, respectively. (See Configuration information)
- Counter bus A can be driven by Unified Channel 23 or by the external shared timer bus(Red-Line Client).
- Each channel has its own timebase, alternative to the counter buses.
- One global prescaler
- One prescaler per channel (CP)
- Timebases shared through the counter buses
- One STAC Client (STAC)
- Control and status bits grouped in separated registers
- Synchronization among timebases
- Shadow FLAG register
- State of the Unified Channels can be frozen for debug purposes

### **34.2.3 Modes of operation**

The Unified Channels can be configured to operate in the following modes:

- General-Purpose Input/Output (GPIO)
- Single-Action Input Capture (SAIC)
- Single-Action Output Compare (SAOC)
- Input Pulse-Width Measurement (IPWM)
- Input Period Measurement (IPM)
- Double-Action Output Compare (DAOC)
- Pulse/Edge Accumulation (PEA)
- Pulse/Edge Counting (PEC)
- Quadrature Decode (QDEC)
- Windowed Programmable Time Accumulation (WPTA)



- Modulus Counter (MC)
- Modulus Counter Buffered (MCB)
- Output Pulse-Width and Frequency Modulation (OPWFM)
- Output Pulse-Width and Frequency Modulation Buffered (OPWFMB)
- Center Aligned Output Pulse Width Modulation with Dead Time Insertion (OPWMC)
- Output Pulse-Width Modulation (OPWM)
- Output Pulse-Width Modulation Buffered (OPWMB)

These modes are described in [Unified Channel modes of operation](#).

Each channel can have a specific set of modes.

## 34.3 Memory map / register definition

### 34.3.1 Memory map

The overall address map organization is shown in the table below.

Whenever an access to either an absent register, absent channel, or reserved address space is performed, the transfer is terminated with an error.

**Table 34-3. eMIOS Memory Map**

| eMIOS[n] Base Address | Description                              |
|-----------------------|------------------------------------------|
| 0x000<br>0x003        | Module Configuration register (EMIOSMCR) |
| 0x004<br>0x007        | Global FLAG register (EMIOSGFLAG)        |
| 0x008<br>0x00B        | Output Update Disable (EMIOSOUDIS)       |
| 0x00C<br>0x00F        | Reserved                                 |
| 0x01F                 | Reserved                                 |
| 0x020<br>0x11F        | Channel 0 to Channel 7                   |
| 0x120<br>0x21F        | Channel 8 to Channel 15                  |

*Table continues on the next page...*

**Table 34-3. eMIOS Memory Map (continued)**

| eMIOS[n] Base Address | Description              |
|-----------------------|--------------------------|
| 0x220<br>0x31F        | Channel 16 to Channel 23 |
| 0x320<br>0x41F        | Channel 24 to Channel 31 |
| 0x420<br>0xFFF        | Reserved                 |

### 34.3.1.1 Unified channel memory map

Addresses of Unified Channel registers are specified as offsets from the channel's base address, otherwise the eMIOS base address is used as reference.

The following table describes the Unified Channel memory map.

**Table 34-4. Unified channel memory map**

| UC[n] Base Address                       | Description                         |
|------------------------------------------|-------------------------------------|
| UC[n]_BASE + 0x000                       | A register (EMIOSA[n])              |
| UC[n]_BASE + 0x004                       | B register (EMIOSB[n])              |
| UC[n]_BASE + 0x008                       | Counter register (EMIOSCNT[n])      |
| UC[n]_BASE + 0x00C                       | Control register (EMIOSC[n])        |
| UC[n]_BASE + 0x010                       | Status register (EMIOSS[n])         |
| UC[n]_BASE + 0x014                       | Alternate A register (EMIOSALTA[n]) |
| UC[n]_BASE + 0x018                       | Control 2 register (EMIOSC2[n])     |
| UC[n]_BASE + 0x01C to UC[n]_BASE + 0x01F | Reserved                            |

## 34.4 Control Memory Map and Registers

All control registers are 32 bits wide. This document illustrates the eMIOS with 24 Unified Channels and 24-bit wide data registers.

### eMIOS memory map

| Address offset (hex) | Register name                                   | Width (in bits) | Access | Reset value | Section/page                |
|----------------------|-------------------------------------------------|-----------------|--------|-------------|-----------------------------|
| 0                    | eMIOS Module Configuration Register (eMIOS_MCR) | 32              | R/W    | 0000_0000h  | <a href="#">34.4.1/1199</a> |

*Table continues on the next page...*

**eMIOS memory map (continued)**

| Address offset (hex) | Register name                                      | Width (in bits) | Access | Reset value | Section/ page               |
|----------------------|----------------------------------------------------|-----------------|--------|-------------|-----------------------------|
| 4                    | eMIOS Global FLAG Register (eMIOS_GFLAG)           | 32              | R      | 0000_0000h  | <a href="#">34.4.2/1201</a> |
| 8                    | eMIOS Output Update Disable register (eMIOS_OUDIS) | 32              | R/W    | 0000_0000h  | <a href="#">34.4.3/1201</a> |

**34.4.1 eMIOS Module Configuration Register (eMIOS\_MCR)**

The MCR register contains Global Control bits for the eMIOS block.

Address: 0h base + 0h offset = 0h

|       |          |      |     |      |     |      |   |   |   |   |    |    |     |    |    |    |
|-------|----------|------|-----|------|-----|------|---|---|---|---|----|----|-----|----|----|----|
| Bit   | 0        | 1    | 2   | 3    | 4   | 5    | 6 | 7 | 8 | 9 | 10 | 11 | 12  | 13 | 14 | 15 |
| R     | Reserved | MDIS | FRZ | GTBE | ETB | GPEN | 0 |   |   |   |    |    | SRV |    |    |    |
| W     |          |      |     |      |     |      |   |   |   |   |    |    |     |    |    |    |
| Reset | 0        | 0    | 0   | 0    | 0   | 0    | 0 | 0 | 0 | 0 | 0  | 0  | 0   | 0  | 0  | 0  |

|       |      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 16   | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | GPEN |    |    |    |    |    |    |    | 0  |    |    |    |    |    |    |    |
| W     |      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**eMIOS\_MCR field descriptions**

| Field         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>Reserved | Reserved.<br><br>This field is reserved.<br>Only write this field with the reset value.                                                                                                                                                                                                                                                                                                                                                                                                     |
| 1<br>MDIS     | Module Disable bit.<br><br>Puts the eMIOS into low power mode. The MDIS bit stops the clock of the block, except the access to registers EMIOSMCR and EMIOSOUDIS.<br><br>0 Clock is running<br>1 Enter low power mode                                                                                                                                                                                                                                                                       |
| 2<br>FRZ      | Freeze bit.<br><br>This bit enables the eMIOS to freeze the registers of the Unified Channels when Debug Mode is requested at MCU level. Each Unified Channel should have FREN bit set in order to enter freeze state. While in Freeze state, the eMIOS continues to operate to allow the MCU access to the Unified Channels registers. The Unified Channel will remain frozen until the FRZ bit is written to zero or the MCU exits Debug mode or the Unified Channel FREN bit is cleared. |

*Table continues on the next page...*

## eMIOS\_MCR field descriptions (continued)

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | 0 Exit freeze state<br>1 Stops Unified Channels operation when in Debug mode and the FREN bit is set in the EMIOSC[n] register                                                                                                                                                                                                                                                                                                                                                 |
| 3<br>GTBE         | Global Time Base Enable bit.<br><br>The GTBE bit is used to export a Global Time Base Enable from the module and provide a method to start time bases of several blocks simultaneously.<br><br><b>NOTE:</b> The Global Time Base Enable input pin controls the internal counters. When asserted, Internal counters are enabled. When negated, Internal counters disabled.<br><br>0 Global Time Base Enable Out signal negated<br>1 Global Time Base Enable Out signal asserted |
| 4<br>ETB          | External Time Base bit<br><br>The ETB bit selects the time base source that drives counter bus[A].<br><br>0 Counter bus[A] assigned to Unified Channel.<br>1 Counter bus[A] assigned to STAC.                                                                                                                                                                                                                                                                                  |
| 5<br>GPREN        | Global Prescaler Enable bit.<br><br>The GPREN bit enables the prescaler counter.<br><br>0 Prescaler disabled (no clock) and prescaler counter is cleared<br>1 Prescaler enabled                                                                                                                                                                                                                                                                                                |
| 6–11<br>Reserved  | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                       |
| 12–15<br>SRV      | Server Time Slot bits.<br><br>The SRV bits select the address of a specific STAC server to which the STAC is assigned.                                                                                                                                                                                                                                                                                                                                                         |
| 16–23<br>GPRE     | Global Prescaler bits.<br><br>The GPRE[ 0:7 ] bits select the clock divider value for the global prescaler, as shown in the following table.<br>GPRE values and corresponding divide ratio values are:<br><br>00000000 1<br>00000001 2<br>00000010 3<br>00000011 4<br>----- -<br>11111110 255<br>11111111 256                                                                                                                                                                  |
| 24–31<br>Reserved | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                       |

### 34.4.2 eMIOS Global FLAG Register (eMIOS\_GFLAG)

GFLAG is a read-only register that groups the FLAG bits from all channels. Each bit relates to one channel.

Address: 0h base + 4h offset = 4h

|       |   |   |   |   |   |   |   |   |         |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---------|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8       | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24    | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   | F23_F16 |   |    |    |    |    |    |    | 0  |    |    |    |    |    |    |    | F7_F0 |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |         |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0       | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0     | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

#### eMIOS\_GFLAG field descriptions

| Field             | Description                                                                                                               |
|-------------------|---------------------------------------------------------------------------------------------------------------------------|
| 0–7<br>Reserved   | Reserved.<br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                      |
| 8–15<br>F23_F16   | Channel [n] mirrored FLAG bit.<br>For Unified Channels these bits are mirrors of the FLAG bits in the EMIOSS[n] register. |
| 16–23<br>Reserved | Reserved.<br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                      |
| 24–31<br>F7_F0    | Channel [n] mirrored FLAG bit.<br>For Unified Channels these bits are mirrors of the FLAG bits in the EMIOSS[n] register. |

### 34.4.3 eMIOS Output Update Disable register (eMIOS\_OUDIS)

Address: 0h base + 8h offset = 8h

|       |   |   |   |   |   |   |   |   |           |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|-----------|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8         | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24      | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   | OU23_OU16 |   |    |    |    |    |    |    | 0  |    |    |    |    |    |    |    | OU7_OU0 |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |           |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0         | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0       | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

#### eMIOS\_OUDIS field descriptions

| Field           | Description                                                                                          |
|-----------------|------------------------------------------------------------------------------------------------------|
| 0–7<br>Reserved | Reserved.<br>This field is reserved.<br>This read-only field is reserved and always has the value 0. |

*Table continues on the next page...*

**eMIOS\_OUDIS field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 8–15<br>OU23_OU16 | <p>Channel [n] Output Update Disable bit.</p> <p>When running MC, MCB or an output mode, values are written to registers A2 and B2. OU[n] bits are used to disable transfers from registers A2 to A1 and B2 to B1. Each bit controls one channel.</p> <p>0 Transfer enabled. Depending on the operation mode, transfer may occur immediately or in the next period. Unless stated otherwise, transfer occurs immediately.</p> <p>1 Transfers disabled.</p> |
| 16–23<br>Reserved | <p>Reserved.</p> <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                        |
| 24–31<br>OU7_OU0  | <p>Channel [n] Output Update Disable bit.</p> <p>When running MC, MCB or an output mode, values are written to registers A2 and B2. OU[n] bits are used to disable transfers from registers A2 to A1 and B2 to B1. Each bit controls one channel.</p> <p>0 Transfer enabled. Depending on the operation mode, transfer may occur immediately or in the next period. Unless stated otherwise, transfer occurs immediately.</p> <p>1 Transfers disabled.</p> |

## 34.5 Unified Channels Memory Map and Registers

The Unified Channel registers are as follows:

**eMIOS memory map**

| Address offset (hex) | Register name                                    | Width (in bits) | Access | Reset value                 | Section/page                |
|----------------------|--------------------------------------------------|-----------------|--------|-----------------------------|-----------------------------|
| 20                   | eMIOS UC A register n (eMIOS_A0)                 | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.1/1205</a> |
| 24                   | eMIOS UC B register n (eMIOS_B0)                 | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.2/1206</a> |
| 28                   | eMIOS CNT register n (eMIOS_CNT0)                | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.3/1207</a> |
| 2C                   | eMIOS UC Control register n (eMIOS_C0)           | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.4/1208</a> |
| 30                   | eMIOS UC Status register n (eMIOS_S0)            | 32              | R/W    | <a href="#">See section</a> | <a href="#">34.5.5/1214</a> |
| 34                   | eMIOS Alternate Address register n (eMIOS_ALTA0) | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.6/1216</a> |
| 40                   | eMIOS UC A register n (eMIOS_A1)                 | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.1/1205</a> |
| 44                   | eMIOS UC B register n (eMIOS_B1)                 | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.2/1206</a> |
| 48                   | eMIOS CNT register n (eMIOS_CNT1)                | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.3/1207</a> |
| 4C                   | eMIOS UC Control register n (eMIOS_C1)           | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.4/1208</a> |
| 50                   | eMIOS UC Status register n (eMIOS_S1)            | 32              | R/W    | <a href="#">See section</a> | <a href="#">34.5.5/1214</a> |
| 54                   | eMIOS Alternate Address register n (eMIOS_ALTA1) | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.6/1216</a> |
| 60                   | eMIOS UC A register n (eMIOS_A2)                 | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.1/1205</a> |

*Table continues on the next page...*

**eMIOS memory map (continued)**

| Address offset (hex) | Register name                                    | Width (in bits) | Access | Reset value                 | Section/ page               |
|----------------------|--------------------------------------------------|-----------------|--------|-----------------------------|-----------------------------|
| 64                   | eMIOS UC B register n (eMIOS_B2)                 | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.2/1206</a> |
| 68                   | eMIOS CNT register n (eMIOS_CNT2)                | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.3/1207</a> |
| 6C                   | eMIOS UC Control register n (eMIOS_C2)           | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.4/1208</a> |
| 70                   | eMIOS UC Status register n (eMIOS_S2)            | 32              | R/W    | <a href="#">See section</a> | <a href="#">34.5.5/1214</a> |
| 74                   | eMIOS Alternate Address register n (eMIOS_ALTA2) | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.6/1216</a> |
| 80                   | eMIOS UC A register n (eMIOS_A3)                 | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.1/1205</a> |
| 84                   | eMIOS UC B register n (eMIOS_B3)                 | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.2/1206</a> |
| 88                   | eMIOS CNT register n (eMIOS_CNT3)                | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.3/1207</a> |
| 8C                   | eMIOS UC Control register n (eMIOS_C3)           | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.4/1208</a> |
| 90                   | eMIOS UC Status register n (eMIOS_S3)            | 32              | R/W    | <a href="#">See section</a> | <a href="#">34.5.5/1214</a> |
| 94                   | eMIOS Alternate Address register n (eMIOS_ALTA3) | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.6/1216</a> |
| A0                   | eMIOS UC A register n (eMIOS_A4)                 | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.1/1205</a> |
| A4                   | eMIOS UC B register n (eMIOS_B4)                 | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.2/1206</a> |
| A8                   | eMIOS CNT register n (eMIOS_CNT4)                | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.3/1207</a> |
| AC                   | eMIOS UC Control register n (eMIOS_C4)           | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.4/1208</a> |
| B0                   | eMIOS UC Status register n (eMIOS_S4)            | 32              | R/W    | <a href="#">See section</a> | <a href="#">34.5.5/1214</a> |
| B4                   | eMIOS Alternate Address register n (eMIOS_ALTA4) | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.6/1216</a> |
| C0                   | eMIOS UC A register n (eMIOS_A5)                 | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.1/1205</a> |
| C4                   | eMIOS UC B register n (eMIOS_B5)                 | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.2/1206</a> |
| C8                   | eMIOS CNT register n (eMIOS_CNT5)                | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.3/1207</a> |
| CC                   | eMIOS UC Control register n (eMIOS_C5)           | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.4/1208</a> |
| D0                   | eMIOS UC Status register n (eMIOS_S5)            | 32              | R/W    | <a href="#">See section</a> | <a href="#">34.5.5/1214</a> |
| D4                   | eMIOS Alternate Address register n (eMIOS_ALTA5) | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.6/1216</a> |
| E0                   | eMIOS UC A register n (eMIOS_A6)                 | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.1/1205</a> |
| E4                   | eMIOS UC B register n (eMIOS_B6)                 | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.2/1206</a> |
| E8                   | eMIOS CNT register n (eMIOS_CNT6)                | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.3/1207</a> |
| EC                   | eMIOS UC Control register n (eMIOS_C6)           | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.4/1208</a> |
| F0                   | eMIOS UC Status register n (eMIOS_S6)            | 32              | R/W    | <a href="#">See section</a> | <a href="#">34.5.5/1214</a> |
| F4                   | eMIOS Alternate Address register n (eMIOS_ALTA6) | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.6/1216</a> |
| 100                  | eMIOS UC A register n (eMIOS_A7)                 | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.1/1205</a> |
| 104                  | eMIOS UC B register n (eMIOS_B7)                 | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.2/1206</a> |
| 108                  | eMIOS CNT register n (eMIOS_CNT7)                | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.3/1207</a> |
| 10C                  | eMIOS UC Control register n (eMIOS_C7)           | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.4/1208</a> |
| 110                  | eMIOS UC Status register n (eMIOS_S7)            | 32              | R/W    | <a href="#">See section</a> | <a href="#">34.5.5/1214</a> |
| 114                  | eMIOS Alternate Address register n (eMIOS_ALTA7) | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.6/1216</a> |
| 220                  | eMIOS UC A register n (eMIOS_A16)                | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.7/1217</a> |
| 224                  | eMIOS UC B register n (eMIOS_B16)                | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.8/1217</a> |
| 228                  | eMIOS CNT register n (eMIOS_CNT16)               | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.9/1218</a> |

*Table continues on the next page...*

**eMIOS memory map (continued)**

| Address offset (hex) | Register name                                     | Width (in bits) | Access | Reset value                 | Section/ page                |
|----------------------|---------------------------------------------------|-----------------|--------|-----------------------------|------------------------------|
| 22C                  | eMIOS UC Control register n (eMIOS_C16)           | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.10/1219</a> |
| 230                  | eMIOS UC Status register n (eMIOS_S16)            | 32              | R/W    | <a href="#">See section</a> | <a href="#">34.5.11/1224</a> |
| 234                  | eMIOS Alternate Address register n (eMIOS_ALTA16) | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.12/1226</a> |
| 240                  | eMIOS UC A register n (eMIOS_A17)                 | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.7/1217</a>  |
| 244                  | eMIOS UC B register n (eMIOS_B17)                 | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.8/1217</a>  |
| 248                  | eMIOS CNT register n (eMIOS_CNT17)                | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.9/1218</a>  |
| 24C                  | eMIOS UC Control register n (eMIOS_C17)           | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.10/1219</a> |
| 250                  | eMIOS UC Status register n (eMIOS_S17)            | 32              | R/W    | <a href="#">See section</a> | <a href="#">34.5.11/1224</a> |
| 254                  | eMIOS Alternate Address register n (eMIOS_ALTA17) | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.12/1226</a> |
| 260                  | eMIOS UC A register n (eMIOS_A18)                 | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.7/1217</a>  |
| 264                  | eMIOS UC B register n (eMIOS_B18)                 | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.8/1217</a>  |
| 268                  | eMIOS CNT register n (eMIOS_CNT18)                | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.9/1218</a>  |
| 26C                  | eMIOS UC Control register n (eMIOS_C18)           | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.10/1219</a> |
| 270                  | eMIOS UC Status register n (eMIOS_S18)            | 32              | R/W    | <a href="#">See section</a> | <a href="#">34.5.11/1224</a> |
| 274                  | eMIOS Alternate Address register n (eMIOS_ALTA18) | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.12/1226</a> |
| 280                  | eMIOS UC A register n (eMIOS_A19)                 | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.7/1217</a>  |
| 284                  | eMIOS UC B register n (eMIOS_B19)                 | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.8/1217</a>  |
| 288                  | eMIOS CNT register n (eMIOS_CNT19)                | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.9/1218</a>  |
| 28C                  | eMIOS UC Control register n (eMIOS_C19)           | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.10/1219</a> |
| 290                  | eMIOS UC Status register n (eMIOS_S19)            | 32              | R/W    | <a href="#">See section</a> | <a href="#">34.5.11/1224</a> |
| 294                  | eMIOS Alternate Address register n (eMIOS_ALTA19) | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.12/1226</a> |
| 2A0                  | eMIOS UC A register n (eMIOS_A20)                 | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.7/1217</a>  |
| 2A4                  | eMIOS UC B register n (eMIOS_B20)                 | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.8/1217</a>  |
| 2A8                  | eMIOS CNT register n (eMIOS_CNT20)                | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.9/1218</a>  |
| 2AC                  | eMIOS UC Control register n (eMIOS_C20)           | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.10/1219</a> |
| 2B0                  | eMIOS UC Status register n (eMIOS_S20)            | 32              | R/W    | <a href="#">See section</a> | <a href="#">34.5.11/1224</a> |
| 2B4                  | eMIOS Alternate Address register n (eMIOS_ALTA20) | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.12/1226</a> |
| 2C0                  | eMIOS UC A register n (eMIOS_A21)                 | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.7/1217</a>  |

*Table continues on the next page...*



**eMIOS memory map (continued)**

| Address offset (hex) | Register name                                     | Width (in bits) | Access | Reset value                 | Section/ page                |
|----------------------|---------------------------------------------------|-----------------|--------|-----------------------------|------------------------------|
| 2C4                  | eMIOS UC B register n (eMIOS_B21)                 | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.8/1217</a>  |
| 2C8                  | eMIOS CNT register n (eMIOS_CNT21)                | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.9/1218</a>  |
| 2CC                  | eMIOS UC Control register n (eMIOS_C21)           | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.10/1219</a> |
| 2D0                  | eMIOS UC Status register n (eMIOS_S21)            | 32              | R/W    | <a href="#">See section</a> | <a href="#">34.5.11/1224</a> |
| 2D4                  | eMIOS Alternate Address register n (eMIOS_ALTA21) | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.12/1226</a> |
| 2E0                  | eMIOS UC A register n (eMIOS_A22)                 | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.7/1217</a>  |
| 2E4                  | eMIOS UC B register n (eMIOS_B22)                 | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.8/1217</a>  |
| 2E8                  | eMIOS CNT register n (eMIOS_CNT22)                | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.9/1218</a>  |
| 2EC                  | eMIOS UC Control register n (eMIOS_C22)           | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.10/1219</a> |
| 2F0                  | eMIOS UC Status register n (eMIOS_S22)            | 32              | R/W    | <a href="#">See section</a> | <a href="#">34.5.11/1224</a> |
| 2F4                  | eMIOS Alternate Address register n (eMIOS_ALTA22) | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.12/1226</a> |
| 300                  | eMIOS UC A register n (eMIOS_A23)                 | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.7/1217</a>  |
| 304                  | eMIOS UC B register n (eMIOS_B23)                 | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.8/1217</a>  |
| 308                  | eMIOS CNT register n (eMIOS_CNT23)                | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.9/1218</a>  |
| 30C                  | eMIOS UC Control register n (eMIOS_C23)           | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.10/1219</a> |
| 310                  | eMIOS UC Status register n (eMIOS_S23)            | 32              | R/W    | <a href="#">See section</a> | <a href="#">34.5.11/1224</a> |
| 314                  | eMIOS Alternate Address register n (eMIOS_ALTA23) | 32              | R/W    | 0000_0000h                  | <a href="#">34.5.12/1226</a> |

**34.5.1 eMIOS UC A register n (eMIOS\_An)**

This section describes the register for channels 0-7. For channels 16-23, see [eMIOS UC A register n \(eMIOS\\_An\)](#).

Address: 0h base + 20h offset + (32d × i), where i=0d to 7d

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   | A |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**eMIOS\_An field descriptions**

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–7<br>Reserved | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                              |
| 8–31<br>A       | Depending on the mode of operation, internal registers A1[ 0:23 ] or A2[ 0:23 ], used for matches and captures, can be assigned to address EMIOS_An. Both A1 and A2 are cleared by reset. The table in the EMIOS_Bn register description summarizes the EMIOS_An writing and reading accesses for all operation modes. Values not listed are reserved. For more information, see the “UC Modes of Operation” section. |

**34.5.2 eMIOS UC B register n (eMIOS\_Bn)**

This section describes the register for channels 0-7. For channels 16-23, see [eMIOS UC B register n \(eMIOS\\_Bn\)](#).

Depending on the mode of operation, internal registers B1[ 0:23 ] or B2[ 0:23 ] can be assigned to address EMIOS\_Bn. Both B1 and B2 are cleared by reset. The table below summarizes the EMIOS\_Bn writing and reading accesses for all operation modes. Values not listed are reserved. For more information see the “UC Modes of Operation” section.

Depending on the channel configuration, it may have an EMIOS\_B register or not. EMIOS\_B register is required for the following modes: DAOC, IPM, IPWM, OPWM, OPWMB, OPWFM , OPWFMB, OPWMC, OPWMCB, MC, MCB, PEA, PEC, WPTA.

The following table summarizes the EMIOS\_An, EMIOS\_Bn, and EMIOS\_ALTAn writing and reading accesses for all operation modes. Values not listed are reserved. For more information, see the “UC Modes of Operation” section.

**Table 34-15. EMIOS\_An, EMIOS\_Bn and EMIOS\_ALTAn values assignment**

| Operation mode    | Register access |      |        |      |           |          |
|-------------------|-----------------|------|--------|------|-----------|----------|
|                   | Write           | Read | Write  | Read | Alt write | Alt read |
| GPIO              | A1, A2          | A1   | B1, B2 | B1   | A2        | A2       |
| SAIC <sup>1</sup> | –               | A2   | B2     | B2   | –         | –        |
| SAOC <sup>1</sup> | A2              | A1   | B2     | B2   | –         | –        |
| IPWM              | –               | A2   | –      | B1   | –         | –        |
| IPM               | –               | A2   | –      | B1   | –         | –        |
| DAOC              | A2              | A1   | B2     | B1   | –         | –        |
| PEA               | A1              | A2   | –      | B1   | –         | –        |
| PEC <sup>1</sup>  | A1              | A1   | B1     | B1   | –         | A2       |
| QDEC <sup>1</sup> | A1              | A1   | B2     | B2   | –         | –        |
| WPTA              | A1              | A1   | B1     | B1   | –         | A2       |

*Table continues on the next page...*

**Table 34-15. EMIOS\_An, EMIOS\_Bn and EMIOS\_ALTAn values assignment (continued)**

| Operation mode   | Register access |      |       |      |           |          |
|------------------|-----------------|------|-------|------|-----------|----------|
|                  | Write           | Read | Write | Read | Alt write | Alt read |
| MC <sup>1</sup>  | A2              | A1   | B2    | B2   | —         | —        |
| OPWFM            | A2              | A1   | B2    | B1   | —         | —        |
| OPWMC            | A2              | A1   | B2    | B1   | —         | —        |
| OPWM             | A2              | A1   | B2    | B1   | —         | —        |
| OPWMT            | A2              | A1   | B2    | B1   | A2        | A2       |
| MCB <sup>1</sup> | A2              | A1   | B2    | B1   | —         | —        |
| OPWFMB           | A2              | A1   | B2    | B1   | —         | —        |
| OPWMCB           | A2              | A1   | B2    | B1   | —         | —        |
| OPWMB            | A2              | A1   | B2    | B1   | —         | —        |

1. In these modes, the register EMIOS\_Bn is not used, but B2 can be accessed.

Address: 0h base + 24h offset + (32d × i), where i=0d to 7d

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   | B |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

### eMIOS\_Bn field descriptions

| Field           | Description                                                                                              |
|-----------------|----------------------------------------------------------------------------------------------------------|
| 0–7<br>Reserved | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 8–31<br>B       | See text above register figure.                                                                          |

1. In these modes, the register EMIOS\_Bn is not used, but B2 can be accessed.

### 34.5.3 eMIOS CNT register n (eMIOS\_CNTn)

This section describes the register for channels 0-7. For channels 16-23, see [eMIOS CNT register n \(eMIOS\\_CNTn\)](#).

When GPIO mode is selected or the channel is frozen, the EMIOS\_CNTn register is read/write. For all others modes, the EMIOS\_CNTn is a read-only register. When entering some operation modes, this register is automatically cleared (refer to the “UC Modes of Operation” section for details).

EMIOS\_CNTn register is required for the following modes:

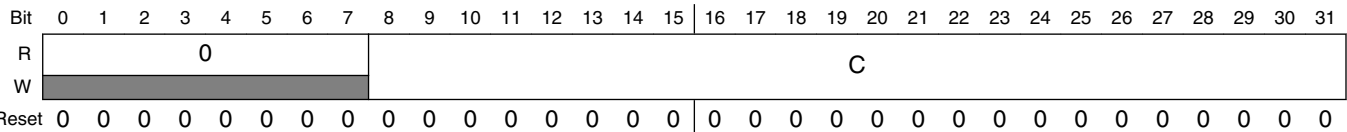
- OPWFM

Unified Channels Memory Map and Registers

- OPWFMB
- OPWMC
- OPWMCB
- PEA
- PEC
- WPTA
- QDEC
- MC
- MCB

For other modes, the internal counter is not used. Therefore, depending on the modes available in a channel, the internal counter may be absent.

Address: 0h base + 28h offset + (32d × i), where i=0d to 7d



eMIOS\_CNTn field descriptions

| Field           | Description                                                                                              |
|-----------------|----------------------------------------------------------------------------------------------------------|
| 0–7<br>Reserved | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 8–31<br>C       | The C field contains the value of the internal counter.                                                  |

34.5.4 eMIOS UC Control register n (eMIOS\_Cn)

This section describes the register for channels 0-7. For channels 16-23, see [eMIOS UC Control register n \(eMIOS\\_Cn\)](#).

The UC Control register gathers several read/write control bits affecting the UC operation conditions.

NOTE

This register does not include a FORCMB bit for Channel Types A and C.

Address: 0h base + 2Ch offset + (32d × i), where i=0d to 7d

|       |      |      |        |   |       |   |        |     |   |   |    |    |    |     |     |    |
|-------|------|------|--------|---|-------|---|--------|-----|---|---|----|----|----|-----|-----|----|
| Bit   | 0    | 1    | 2      | 3 | 4     | 5 | 6      | 7   | 8 | 9 | 10 | 11 | 12 | 13  | 14  | 15 |
| R     |      |      |        |   |       |   |        |     | 0 |   |    |    |    |     |     | 0  |
| W     | FREN | ODIS | ODISSL |   | UCPRE |   | UCPREN | DMA |   |   | IF |    |    | FCK | FEN |    |
| Reset | 0    | 0    | 0      | 0 | 0     | 0 | 0      | 0   | 0 | 0 | 0  | 0  | 0  | 0   | 0   | 0  |

|       |    |    |        |        |    |     |    |       |       |    |    |    |    |    |    |    |
|-------|----|----|--------|--------|----|-----|----|-------|-------|----|----|----|----|----|----|----|
| Bit   | 16 | 17 | 18     | 19     | 20 | 21  | 22 | 23    | 24    | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0  |    |        |        | 0  |     |    |       |       |    |    |    |    |    |    |    |
| W     |    |    | FORCMA | FORCMB |    | BSL |    | EDSEL | EDPOL |    |    |    |    |    |    |    |
| Reset | 0  | 0  | 0      | 0      | 0  | 0   | 0  | 0     | 0     | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

### eMIOS\_Cn field descriptions

| Field         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |               |              |    |                        |    |                        |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|--------------|----|------------------------|----|------------------------|
| 0<br>FREN     | <p>Freeze Enable bit.</p> <p>The FREN bit, if set and validated by FRZ bit in EMIOS_MCR register allows the channel to enter freeze state, freezing all registers values when in debug mode and allowing the MCU to perform debug functions.</p> <p>0 Normal operation<br/>1 Freeze UC registers values</p>                                                                                                                                                                                                                                                                     |               |              |    |                        |    |                        |
| 1<br>ODIS     | <p>Output Disable bit.</p> <p>The ODIS bit allows disabling the output pin when running any of the output modes with the exception of GPIO mode.</p> <p>0 The output pin operates normally<br/>1 If the selected Output Disable Input signal is asserted, the output pin goes to EDPOL for OPWFMB and OPWMB modes and to the complement of EDPOL for other output modes, but the Unified Channel continues to operate normally, i.e., it continues to produce FLAG and matches. When the selected Output Disable Input signal is negated, the output pin operates normally.</p> |               |              |    |                        |    |                        |
| 2–3<br>ODISSL | <p>Output Disable select bits.</p> <p>The ODISSL[ 0:1 ] bits select one of the four output disable input signals, as shown in the following table.</p> <p><b>Table 34-29. UC ODISSL selection</b></p> <table> <tr> <th>ODISSL[ 0:1 ]</th><th>input signal</th></tr> <tr> <td>00</td><td>Output Disable Input 0</td></tr> <tr> <td>01</td><td>Output Disable Input 1</td></tr> </table>                                                                                                                                                                                          | ODISSL[ 0:1 ] | input signal | 00 | Output Disable Input 0 | 01 | Output Disable Input 1 |
| ODISSL[ 0:1 ] | input signal                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |               |              |    |                        |    |                        |
| 00            | Output Disable Input 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |               |              |    |                        |    |                        |
| 01            | Output Disable Input 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |               |              |    |                        |    |                        |

Table continues on the next page...

## eMIOS\_Cn field descriptions (continued)

| Field                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                       |                                                  |      |                        |      |                        |      |    |    |   |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|--------------------------------------------------|------|------------------------|------|------------------------|------|----|----|---|
|                       | <div>Table 34-29. UC ODISSL selection<br/>(continued)</div> <table><tr><th>ODISSL[ 0:1 ]</th><th>input signal</th></tr><tr><td>10</td><td>Output Disable Input 2</td></tr><tr><td>11</td><td>Output Disable Input 3</td></tr></table>                                                                                                                                                                                                                                                                                                                                                                                               | ODISSL[ 0:1 ]         | input signal                                     | 10   | Output Disable Input 2 | 11   | Output Disable Input 3 |      |    |    |   |
| ODISSL[ 0:1 ]         | input signal                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                       |                                                  |      |                        |      |                        |      |    |    |   |
| 10                    | Output Disable Input 2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                       |                                                  |      |                        |      |                        |      |    |    |   |
| 11                    | Output Disable Input 3                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                       |                                                  |      |                        |      |                        |      |    |    |   |
| 4–5<br>UCPRE          | <div>Prescaler bits.</div> <div>The UCPRE[ 0:1 ] bits select the clock divider value for the internal prescaler of Unified Channel, as shown in the following table.</div> <div>Table 34-28. UC Internal Prescaler clock divider</div> <table><tr><th>UCPRE[ 0:1]</th><th>Divide ratio</th></tr><tr><td>00</td><td>1</td></tr><tr><td>01</td><td>2</td></tr><tr><td>10</td><td>3</td></tr><tr><td>11</td><td>4</td></tr></table> <div>NOTE: The UCPRE bits are mirrored in the two LSBs of the UCEXTPRE field in register EMIOS_Cn. Any write operations to UCPRE bits will affect the respective bits at EMIOS_C2n register.</div> | UCPRE[ 0:1]           | Divide ratio                                     | 00   | 1                      | 01   | 2                      | 10   | 3  | 11 | 4 |
| UCPRE[ 0:1]           | Divide ratio                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                       |                                                  |      |                        |      |                        |      |    |    |   |
| 00                    | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                       |                                                  |      |                        |      |                        |      |    |    |   |
| 01                    | 2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                       |                                                  |      |                        |      |                        |      |    |    |   |
| 10                    | 3                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                       |                                                  |      |                        |      |                        |      |    |    |   |
| 11                    | 4                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                       |                                                  |      |                        |      |                        |      |    |    |   |
| 6<br>UCPREN           | <div>Prescaler Enable bit.</div> <div>The UCPREN bit enables the prescaler counter.</div> <div>0 Prescaler disabled (no clock)</div> <div>1 Prescaler enabled</div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                       |                                                  |      |                        |      |                        |      |    |    |   |
| 7<br>DMA              | <div>Direct Memory Access bit.</div> <div>The DMA bit selects if FLAG generation will be used as an interrupt or as a DMA request.</div> <div>0 Flag/overflow assigned to Interrupt request.</div> <div>1 Flag/overflow assigned to DMA request.</div>                                                                                                                                                                                                                                                                                                                                                                              |                       |                                                  |      |                        |      |                        |      |    |    |   |
| 8<br>Reserved         | <div>Reserved.</div> <div>This field is reserved.</div> <div>This read-only field is reserved and always has the value 0.</div>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                       |                                                  |      |                        |      |                        |      |    |    |   |
| 9–12<br>IF            | <div>Input Filter bits.</div> <div>The IF[ 0:3 ] bits control the programmable input filter, selecting the minimum input pulse width that can pass through the filter, as shown in the following table. For output modes, these bits have no meaning.</div> <div>Table 34-27. UC Input Filter bits</div> <table><tr><th>IF[ 0:3] <sup>1</sup></th><th>Minimum input Pulse width [filter clock periods]</th></tr><tr><td>0000</td><td>bypassed<sup>2</sup></td></tr><tr><td>0001</td><td>02</td></tr><tr><td>0010</td><td>04</td></tr></table>                                                                                       | IF[ 0:3] <sup>1</sup> | Minimum input Pulse width [filter clock periods] | 0000 | bypassed <sup>2</sup>  | 0001 | 02                     | 0010 | 04 |    |   |
| IF[ 0:3] <sup>1</sup> | Minimum input Pulse width [filter clock periods]                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                       |                                                  |      |                        |      |                        |      |    |    |   |
| 0000                  | bypassed <sup>2</sup>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                       |                                                  |      |                        |      |                        |      |    |    |   |
| 0001                  | 02                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                       |                                                  |      |                        |      |                        |      |    |    |   |
| 0010                  | 04                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                       |                                                  |      |                        |      |                        |      |    |    |   |

Table continues on the next page...

**eMIOS\_Cn field descriptions (continued)**

| Field                                                                                           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                                  |                                                  |      |    |      |    |            |          |
|-------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|--------------------------------------------------|------|----|------|----|------------|----------|
|                                                                                                 | <b>Table 34-27. UC Input Filter bits (continued)</b>                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                  |                                                  |      |    |      |    |            |          |
|                                                                                                 | <table><tr><th>IF[ 0:3] <sup>1</sup></th><th>Minimum input Pulse width [filter clock periods]</th></tr><tr><td>0100</td><td>08</td></tr><tr><td>1000</td><td>16</td></tr><tr><td>All others</td><td>Reserved</td></tr></table>                                                                                                                                                                                                                                            | IF[ 0:3] <sup>1</sup>                            | Minimum input Pulse width [filter clock periods] | 0100 | 08 | 1000 | 16 | All others | Reserved |
|                                                                                                 | IF[ 0:3] <sup>1</sup>                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Minimum input Pulse width [filter clock periods] |                                                  |      |    |      |    |            |          |
|                                                                                                 | 0100                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 08                                               |                                                  |      |    |      |    |            |          |
|                                                                                                 | 1000                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 16                                               |                                                  |      |    |      |    |            |          |
|                                                                                                 | All others                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Reserved                                         |                                                  |      |    |      |    |            |          |
| 1. Filter latency (the difference in time between the input and the response) is 3 clock edges. |                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                                  |                                                  |      |    |      |    |            |          |
| 2. The input signal is synchronized before arriving to the digital filter.                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                                  |                                                  |      |    |      |    |            |          |
|                                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                                  |                                                  |      |    |      |    |            |          |
| 13<br>FCK                                                                                       | Filter Clock select bit.<br><br>The FCK bit selects the clock source for the programmable input filter.<br><br>0 Prescaled clock<br>1 eMIOS module clock                                                                                                                                                                                                                                                                                                                  |                                                  |                                                  |      |    |      |    |            |          |
| 14<br>FEN                                                                                       | FLAG Enable bit.<br><br>The FEN bit allows the Unified Channel FLAG bit to generate an interrupt signal or a DMA request signal (the type of signal to be generated is defined by the DMA bit).<br><br>0 Disable (FLAG does not generate an interrupt request)<br>1 Enable (FLAG generates an interrupt request)                                                                                                                                                          |                                                  |                                                  |      |    |      |    |            |          |
| 15–17<br>Reserved                                                                               | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                  |                                                  |                                                  |      |    |      |    |            |          |
| 18<br>FORCMA                                                                                    | Force Match A bit.<br><br>For output modes, the FORCMA bit is equivalent to a successful comparison on comparator A (except that the FLAG bit is not set). This bit is cleared by reset and is always read as zero. This bit is valid for every output operation mode which uses comparator A, otherwise it has no effect.<br><br>For input modes, the FORCMA bit is not used and writing to it has no effect.<br><br>0 Has no effect.<br>1 Force a match at comparator A |                                                  |                                                  |      |    |      |    |            |          |
| 19<br>FORCMB                                                                                    | Force Match B bit.<br><br>For output modes, the FORCMB bit is equivalent to a successful comparison on comparator B (except that the FLAG bit is not set). This bit is cleared by reset and is always read as zero. This bit is valid for every output operation mode which uses comparator B, otherwise it has no effect.<br><br>0 Has not effect<br>1 Force a match at comparator B                                                                                     |                                                  |                                                  |      |    |      |    |            |          |
| 20<br>Reserved                                                                                  | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                  |                                                  |                                                  |      |    |      |    |            |          |
| 21–22<br>BSL                                                                                    | Bus Select bits.                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                  |                                                  |      |    |      |    |            |          |

*Table continues on the next page...*

## eMIOS\_Cn field descriptions (continued)

| Field       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |           |                           |    |                              |    |                                 |                                  |                                   |                                   |    |          |    |                                |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|---------------------------|----|------------------------------|----|---------------------------------|----------------------------------|-----------------------------------|-----------------------------------|----|----------|----|--------------------------------|
|             | <p>The BSL[ 0:1 ] bits are used to select either one of the counter buses or the internal counter to be used by the Unified Channel , as shown in the following tables.</p> <p style="text-align: center;"><b>Table 34-26. UC BSL bits</b></p> <table> <tr> <th>BSL[ 0:1]</th><th>Selected bus<sup>1</sup></th></tr> <tr> <td>00</td><td>All channels: counter bus[A]</td></tr> <tr> <td rowspan="4">01</td><td>Channels 0 to 7: counter bus[B]</td></tr> <tr> <td>Channels 8 to 15: counter bus[C]</td></tr> <tr> <td>Channels 16 to 23: counter bus[D]</td></tr> <tr> <td>Channels 24 to 31: counter bus[E]</td></tr> <tr> <td>10</td><td>Reserved</td></tr> <tr> <td>11</td><td>All channels: internal counter</td></tr> </table> <p>1. Not all devices support all counter buses. See the Device Configuration chapter for details.</p>                                                                                 | BSL[ 0:1] | Selected bus <sup>1</sup> | 00 | All channels: counter bus[A] | 01 | Channels 0 to 7: counter bus[B] | Channels 8 to 15: counter bus[C] | Channels 16 to 23: counter bus[D] | Channels 24 to 31: counter bus[E] | 10 | Reserved | 11 | All channels: internal counter |
| BSL[ 0:1]   | Selected bus <sup>1</sup>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |           |                           |    |                              |    |                                 |                                  |                                   |                                   |    |          |    |                                |
| 00          | All channels: counter bus[A]                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |           |                           |    |                              |    |                                 |                                  |                                   |                                   |    |          |    |                                |
| 01          | Channels 0 to 7: counter bus[B]                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |           |                           |    |                              |    |                                 |                                  |                                   |                                   |    |          |    |                                |
|             | Channels 8 to 15: counter bus[C]                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |           |                           |    |                              |    |                                 |                                  |                                   |                                   |    |          |    |                                |
|             | Channels 16 to 23: counter bus[D]                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |           |                           |    |                              |    |                                 |                                  |                                   |                                   |    |          |    |                                |
|             | Channels 24 to 31: counter bus[E]                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |           |                           |    |                              |    |                                 |                                  |                                   |                                   |    |          |    |                                |
| 10          | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |           |                           |    |                              |    |                                 |                                  |                                   |                                   |    |          |    |                                |
| 11          | All channels: internal counter                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |           |                           |    |                              |    |                                 |                                  |                                   |                                   |    |          |    |                                |
| 23<br>EDSEL | <p>Edge Selection bit.</p> <p>For input modes, the EDSEL bit selects whether the internal counter is triggered by both edges of a pulse or just by a single edge as defined by the EDPOL bit . When not shown in the mode of operation description, this bit has no effect.</p> <p>0 – Single edge triggering defined by the EDPOL bit</p> <p>1 – Both edges triggering{mts_ipi_slv_saic}</p> <p>For GPIO in mode, the EDSEL bit selects if a FLAG can be generated.</p> <p>0 – A FLAG is generated as defined by the EDPOL bit</p> <p>1 – No FLAG is generated</p> <p>For SAOC mode, the EDSEL bit selects the behavior of the output flip-flop at each match.</p> <p>0 – The EDPOL value is transferred to the output flip-flop</p> <p>1 – The output flip-flop is toggled</p>                                                                                                                                            |           |                           |    |                              |    |                                 |                                  |                                   |                                   |    |          |    |                                |
| 24<br>EDPOL | <p>Edge Polarity bit.</p> <p>For input modes (except QDEC mode), the EDPOL bit asserts which edge triggers either the internal counter or an input capture or a FLAG . When not shown in the mode of operation description, this bit has no effect.</p> <p>0 – Trigger on a falling edge</p> <p>1 – Trigger on a rising edge</p> <p>For QDEC(MODE[0] cleared), the EDPOL bit selects the count direction according to <i>direction</i> signal (UC[n] input).</p> <p>1 - counts up when UC[n] is asserted</p> <p>0 - counts down when UC[n] is asserted</p> <p><b>NOTE:</b> UC[n-1] EDPOL bit selects which edge clocks the internal counter of UC[n]</p> <p>For QDEC(MODE[0] cleared), the EDPOL bit selects the count direction according to the phase difference.</p> <p>1 - Internal counter increments if phase_A is ahead phase_B signal</p> <p>0 - Internal counter decrements if phase_A is ahead phase_B signal</p> |           |                           |    |                              |    |                                 |                                  |                                   |                                   |    |          |    |                                |

Table continues on the next page...



**eMIOS\_Cn field descriptions (continued)**

| Field                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |                      |                                                                          |                      |                                                                         |                      |                                                  |                      |                                                    |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|--------------------------------|---------|-------------------------------------------|---------|--------------------------------------------|---------|-----------------------------|---------|------------------------------|---------|-------------------------------|---------|--------------------------|---------|---------------------------------------------------------|---------|------------------------------------------------------------|---------|--------------------------------------|---------|---------------------------------------|---------|----------------------------------|---------|-----------------------------------|---------|-----------------------------------------------------------|---------|-----------------------------------------------------------|---------|-----------------------------------------|---------|----------|----------------------|--------------------------------------------------------|---------|------------------------------------------------------|----------------------|-----------------------------------|----------------------|----------------------------------------------------------------|----------------------|------------------------------------------------------------------|----------------------|--------------------------------------------------------------------------|----------------------|-------------------------------------------------------------------------|----------------------|--------------------------------------------------|----------------------|----------------------------------------------------|
|                      | <p><b>NOTE:</b> In order to operate properly, EDPOL bit must contain the same value in UC[n] and UC[n-1]</p> <p>For output modes, the EDPOL bit is used to select the logic level on the output pin .</p> <p>0 – A match on comparator A clears the output flip-flop, while a match on comparator B sets it</p> <p>1 – A match on comparator A sets the output flip-flop, while a match on comparator B clears it</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |                      |                                                                          |                      |                                                                         |                      |                                                  |                      |                                                    |
| 25–31<br>MODE        | <p>Mode selection bits.</p> <p>The MODE bits select the mode of operation of the Unified Channel, as shown in the following table. Values not listed are reserved.</p> <p><b>NOTE:</b> If a reserved value is written to the mode bit field, the results are unpredictable.</p> <p><b>Table 34-25. UC MODE bits</b></p> <table> <tr> <th>MODE[ 0:6]</th><th>Mode of operation<sup>1</sup></th></tr> <tr><td>0000000</td><td>General purpose Input/Output mode (input)</td></tr> <tr><td>0000001</td><td>General purpose Input/Output mode (output)</td></tr> <tr><td>0000010</td><td>Single Action Input Capture</td></tr> <tr><td>0000011</td><td>Single Action Output Compare</td></tr> <tr><td>0000100</td><td>Input Pulse Width Measurement</td></tr> <tr><td>0000101</td><td>Input Period Measurement</td></tr> <tr><td>0000110</td><td>Double Action Output compare (with FLAG set on B match)</td></tr> <tr><td>0000111</td><td>Double Action Output compare (with FLAG set on both match)</td></tr> <tr><td>0001000</td><td>Pulse/Edge Accumulation (continuous)</td></tr> <tr><td>0001001</td><td>Pulse/Edge Accumulation (single shot)</td></tr> <tr><td>0001010</td><td>Pulse/Edge Counting (continuous)</td></tr> <tr><td>0001011</td><td>Pulse/Edge Counting (single shot)</td></tr> <tr><td>0001100</td><td>Quadrature Decode (for count and direction encoders type)</td></tr> <tr><td>0001101</td><td>Quadrature Decode (for phase_A and phase_B encoders type)</td></tr> <tr><td>0001110</td><td>Windowed Programmable Time Accumulation</td></tr> <tr><td>0001111</td><td>Reserved</td></tr> <tr><td>001000b<sup>2</sup></td><td>Modulus Counter (Up counter with clear on match start)</td></tr> <tr><td>001001b</td><td>Modulus Counter (Up counter with clear on match end)</td></tr> <tr><td>00101bb<sup>2</sup></td><td>Modulus Counter (Up/Down counter)</td></tr> <tr><td>00110b0<sup>2</sup></td><td>Output Pulse Width and Frequency Modulation (immediate update)</td></tr> <tr><td>00110b1<sup>2</sup></td><td>Output Pulse Width and Frequency Modulation (next period update)</td></tr> <tr><td>00111b0<sup>2</sup></td><td>Center Aligned Output Pulse Width Modulation (with trail edge dead-time)</td></tr> <tr><td>00111b1<sup>2</sup></td><td>Center Aligned Output Pulse Width Modulation (with lead edge dead-time)</td></tr> <tr><td>01000b0<sup>2</sup></td><td>Output Pulse Width Modulation (immediate update)</td></tr> <tr><td>01000b1<sup>2</sup></td><td>Output Pulse Width Modulation (next period update)</td></tr> </table> | MODE[ 0:6] | Mode of operation <sup>1</sup> | 0000000 | General purpose Input/Output mode (input) | 0000001 | General purpose Input/Output mode (output) | 0000010 | Single Action Input Capture | 0000011 | Single Action Output Compare | 0000100 | Input Pulse Width Measurement | 0000101 | Input Period Measurement | 0000110 | Double Action Output compare (with FLAG set on B match) | 0000111 | Double Action Output compare (with FLAG set on both match) | 0001000 | Pulse/Edge Accumulation (continuous) | 0001001 | Pulse/Edge Accumulation (single shot) | 0001010 | Pulse/Edge Counting (continuous) | 0001011 | Pulse/Edge Counting (single shot) | 0001100 | Quadrature Decode (for count and direction encoders type) | 0001101 | Quadrature Decode (for phase_A and phase_B encoders type) | 0001110 | Windowed Programmable Time Accumulation | 0001111 | Reserved | 001000b <sup>2</sup> | Modulus Counter (Up counter with clear on match start) | 001001b | Modulus Counter (Up counter with clear on match end) | 00101bb <sup>2</sup> | Modulus Counter (Up/Down counter) | 00110b0 <sup>2</sup> | Output Pulse Width and Frequency Modulation (immediate update) | 00110b1 <sup>2</sup> | Output Pulse Width and Frequency Modulation (next period update) | 00111b0 <sup>2</sup> | Center Aligned Output Pulse Width Modulation (with trail edge dead-time) | 00111b1 <sup>2</sup> | Center Aligned Output Pulse Width Modulation (with lead edge dead-time) | 01000b0 <sup>2</sup> | Output Pulse Width Modulation (immediate update) | 01000b1 <sup>2</sup> | Output Pulse Width Modulation (next period update) |
| MODE[ 0:6]           | Mode of operation <sup>1</sup>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |                      |                                                                          |                      |                                                                         |                      |                                                  |                      |                                                    |
| 0000000              | General purpose Input/Output mode (input)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |                      |                                                                          |                      |                                                                         |                      |                                                  |                      |                                                    |
| 0000001              | General purpose Input/Output mode (output)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |                      |                                                                          |                      |                                                                         |                      |                                                  |                      |                                                    |
| 0000010              | Single Action Input Capture                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |                      |                                                                          |                      |                                                                         |                      |                                                  |                      |                                                    |
| 0000011              | Single Action Output Compare                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |                      |                                                                          |                      |                                                                         |                      |                                                  |                      |                                                    |
| 0000100              | Input Pulse Width Measurement                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |                      |                                                                          |                      |                                                                         |                      |                                                  |                      |                                                    |
| 0000101              | Input Period Measurement                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |                      |                                                                          |                      |                                                                         |                      |                                                  |                      |                                                    |
| 0000110              | Double Action Output compare (with FLAG set on B match)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |                      |                                                                          |                      |                                                                         |                      |                                                  |                      |                                                    |
| 0000111              | Double Action Output compare (with FLAG set on both match)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |                      |                                                                          |                      |                                                                         |                      |                                                  |                      |                                                    |
| 0001000              | Pulse/Edge Accumulation (continuous)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |                      |                                                                          |                      |                                                                         |                      |                                                  |                      |                                                    |
| 0001001              | Pulse/Edge Accumulation (single shot)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |                      |                                                                          |                      |                                                                         |                      |                                                  |                      |                                                    |
| 0001010              | Pulse/Edge Counting (continuous)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |                      |                                                                          |                      |                                                                         |                      |                                                  |                      |                                                    |
| 0001011              | Pulse/Edge Counting (single shot)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |                      |                                                                          |                      |                                                                         |                      |                                                  |                      |                                                    |
| 0001100              | Quadrature Decode (for count and direction encoders type)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |                      |                                                                          |                      |                                                                         |                      |                                                  |                      |                                                    |
| 0001101              | Quadrature Decode (for phase_A and phase_B encoders type)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |                      |                                                                          |                      |                                                                         |                      |                                                  |                      |                                                    |
| 0001110              | Windowed Programmable Time Accumulation                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |                      |                                                                          |                      |                                                                         |                      |                                                  |                      |                                                    |
| 0001111              | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |                      |                                                                          |                      |                                                                         |                      |                                                  |                      |                                                    |
| 001000b <sup>2</sup> | Modulus Counter (Up counter with clear on match start)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |                      |                                                                          |                      |                                                                         |                      |                                                  |                      |                                                    |
| 001001b              | Modulus Counter (Up counter with clear on match end)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |                      |                                                                          |                      |                                                                         |                      |                                                  |                      |                                                    |
| 00101bb <sup>2</sup> | Modulus Counter (Up/Down counter)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |                      |                                                                          |                      |                                                                         |                      |                                                  |                      |                                                    |
| 00110b0 <sup>2</sup> | Output Pulse Width and Frequency Modulation (immediate update)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |                      |                                                                          |                      |                                                                         |                      |                                                  |                      |                                                    |
| 00110b1 <sup>2</sup> | Output Pulse Width and Frequency Modulation (next period update)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |                      |                                                                          |                      |                                                                         |                      |                                                  |                      |                                                    |
| 00111b0 <sup>2</sup> | Center Aligned Output Pulse Width Modulation (with trail edge dead-time)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |                      |                                                                          |                      |                                                                         |                      |                                                  |                      |                                                    |
| 00111b1 <sup>2</sup> | Center Aligned Output Pulse Width Modulation (with lead edge dead-time)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |                      |                                                                          |                      |                                                                         |                      |                                                  |                      |                                                    |
| 01000b0 <sup>2</sup> | Output Pulse Width Modulation (immediate update)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |                      |                                                                          |                      |                                                                         |                      |                                                  |                      |                                                    |
| 01000b1 <sup>2</sup> | Output Pulse Width Modulation (next period update)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |                      |                                                                          |                      |                                                                         |                      |                                                  |                      |                                                    |

Table continues on the next page...

**eMIOS\_Cn field descriptions (continued)**

| Field | Description                                                                                                                                                                                              |                                                                                   |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|
|       | <b>Table 34-25. UC MODE bits (continued)</b>                                                                                                                                                             |                                                                                   |
|       | <b>MODE[ 0:6]</b>                                                                                                                                                                                        | <b>Mode of operation<sup>1</sup></b>                                              |
|       | 0100100 through 0100101                                                                                                                                                                                  | Reserved                                                                          |
|       | 0100110                                                                                                                                                                                                  | Output Pulse Width Modulation with Trigger                                        |
|       | 0100111 through 1001111                                                                                                                                                                                  | Reserved                                                                          |
|       | 101000b <sup>2</sup>                                                                                                                                                                                     | Modulus Counter Buffered (Up counter)                                             |
|       | 101001b <sup>2</sup>                                                                                                                                                                                     | Reserved                                                                          |
|       | 10101bb <sup>2</sup>                                                                                                                                                                                     | Modulus Counter Buffered (Up/Down counter)                                        |
|       | 10110b0 <sup>2</sup>                                                                                                                                                                                     | Output Pulse Width and Frequency Modulation Buffered                              |
|       | 10110b1 <sup>2</sup>                                                                                                                                                                                     | Reserved                                                                          |
|       | 10111b0 <sup>2</sup>                                                                                                                                                                                     | Center Aligned Output Pulse Width Modulation Buffered (with trail edge dead-time) |
|       | 10111b1 <sup>2</sup>                                                                                                                                                                                     | Center Aligned Output Pulse Width Modulation Buffered (with lead edge dead-time)  |
|       | 11000b0 <sup>2</sup>                                                                                                                                                                                     | Output Pulse Width Modulation Buffered                                            |
|       | 1100001 through 1111111                                                                                                                                                                                  | Reserved                                                                          |
|       | 1. Not all modes of operation are supported. See the configuration information for details.<br>2. b = adjust parameters for the mode of operation. Refer to “UC Modes of Operation” section for details. |                                                                                   |

1. Filter latency (the difference in time between the input and the response) is 3 clock edges.
2. The input signal is synchronized before arriving to the digital filter.
3. Not all devices support all counter buses. See the Device Configuration chapter for details.
4. Not all modes of operation are supported. See the configuration information for details.
5. b = adjust parameters for the mode of operation. Refer to “UC Modes of Operation” section for details.

**34.5.5 eMIOS UC Status register n (eMIOS\_Sn)**

This section describes the register for channels 0-7. For channels 16-23, see [eMIOS UC Status register n \(eMIOS\\_Sn\)](#).

The UC Status register EMIOS\_Sn gathers bits reflecting the status of the UC input/output signals and the overflow condition of the internal counter.

Address: 0h base + 30h offset + (32d × i), where i=0d to 7d

|       |      |    |    |    |    |    |    |    |    |    |    |    |    |      |       |      |
|-------|------|----|----|----|----|----|----|----|----|----|----|----|----|------|-------|------|
| Bit   | 0    | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13   | 14    | 15   |
| R     | OVR  | 0  |    |    |    |    |    |    |    |    |    |    |    |      |       |      |
| W     | w1c  |    |    |    |    |    |    |    |    |    |    |    |    |      |       |      |
| Reset | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0     | 0    |
| Bit   | 16   | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29   | 30    | 31   |
| R     | OVFL | 0  |    |    |    |    |    |    |    |    |    |    |    | UCIN | UCOUT | FLAG |
| W     | w1c  |    |    |    |    |    |    |    |    |    |    |    |    |      |       | w1c  |
| Reset | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | *    | 0     | 0    |

\* Notes:

- UCIN field: Reset value is undefined.

**eMIOS\_*Sn* field descriptions**

| Field             | Description                                                                                                                                                                                                                                                                                   |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>OVR          | <p>Overflow bit.</p> <p>The OVR bit indicates that FLAG generation occurred when the FLAG bit was already set .</p> <p>The OVR bit can be cleared either by clearing the FLAG bit or by writing a 1 to the OVR bit.</p> <p>0   Overflow has not occurred</p> <p>1   Overflow has occurred</p> |
| 1–15<br>Reserved  | <p>Reserved.</p> <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                           |
| 16<br>OVFL        | <p>Overflow bit.</p> <p>The OVFL bit indicates that an overflow has occurred in the internal counter.</p> <p>The OVFL bit must be cleared by software writing a 1 to it.</p> <p>0   No overflow</p> <p>1   An overflow had occurred</p>                                                       |
| 17–28<br>Reserved | <p>Reserved.</p> <p>This field is reserved.</p>                                                                                                                                                                                                                                               |

*Table continues on the next page...*

**eMIOS\_Sn field descriptions (continued)**

| Field       | Description                                                                                                                                                                                                       |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|             | This read-only field is reserved and always has the value 0.                                                                                                                                                      |
| 29<br>UCIN  | Unified Channel Input pin bit.<br>The UCIN bit reflects the input pin state after being filtered and synchronized.                                                                                                |
| 30<br>UCOUT | Unified Channel Output pin bit.<br>The UCOUT bit reflects the output pin state .                                                                                                                                  |
| 31<br>FLAG  | FLAG bit.<br>The FLAG bit is set when an input capture or a match event in the comparators occurred.<br>The FLAG bit must be cleared by writing a 1 to it.<br><br>0 FLAG cleared<br>1 FLAG set event has occurred |

**34.5.6 eMIOS Alternate Address register n (eMIOS\_ALTAn)**

This section describes the register for channels 0-7. For channels 16-23, see [eMIOS Alternate Address register n \(eMIOS\\_ALTAn\)](#).

The eMIOS\_ALTAn register provides an alternate address to access A2 channel registers in restricted modes (GPIO, PEC, WPTA )only. In modes where the function of eMIOS\_ALTAn register is not specified, any write access is ignored and any read access returns unspecified data. If eMIOS\_An register is used along with eMIOS\_ALTAn, both A1 and A2 registers can be accessed in these modes. The table in the eMIOS\_Bn register's description summarizes the EMIOS\_ALTAn writing and reading accesses for all EMIOS\_ALTAn operation modes. See the “General purpose Input/Output mode (GPIO) Mode” section, the “Pulse/Edge Counting (PEC) Mode” section, the “Windowed Programmable Time Accumulation (WPTA) Mode” section for a more detailed description of the use of eMIOS\_ALTAn register.

Address: 0h base + 34h offset + (32d × i), where i=0d to 7d

|       |   |   |   |   |   |   |   |   |      |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|------|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8    | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   | ALTA |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |      |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0    | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**eMIOS\_ALTAn field descriptions**

| Field           | Description                                                                                          |
|-----------------|------------------------------------------------------------------------------------------------------|
| 0–7<br>Reserved | Reserved.<br>This field is reserved.<br>This read-only field is reserved and always has the value 0. |

*Table continues on the next page...*

**eMIOS\_ALTAn field descriptions (continued)**

| Field        | Description                     |
|--------------|---------------------------------|
| 8–31<br>ALTA | See text above register figure. |

**34.5.7 eMIOS UC A register n (eMIOS\_An)**

This section describes the register for channels 16-23. For channels 0-7, see [eMIOS UC A register n \(eMIOS\\_An\)](#).

Address: 0h base + 220h offset + (32d × i), where i=0d to 7d

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   | A |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**eMIOS\_An field descriptions**

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–7<br>Reserved | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                              |
| 8–31<br>A       | Depending on the mode of operation, internal registers A1[ 0:23 ] or A2[ 0:23 ], used for matches and captures, can be assigned to address EMIOs_An. Both A1 and A2 are cleared by reset. The table in the EMIOs_Bn register description summarizes the EMIOs_An writing and reading accesses for all operation modes. Values not listed are reserved. For more information, see the “UC Modes of Operation” section. |

**34.5.8 eMIOS UC B register n (eMIOS\_Bn)**

This section describes the register for channels 16-23. In all other respects, this register is identical to the register for channels 0-7. See that register's description: [eMIOS UC B register n \(eMIOS\\_Bn\)](#).

Address: 0h base + 224h offset + (32d × i), where i=0d to 7d

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   | B |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**eMIOS\_Bn field descriptions**

| Field           | Description                                                                                              |
|-----------------|----------------------------------------------------------------------------------------------------------|
| 0–7<br>Reserved | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 8–31<br>B       | See text above register figure.                                                                          |

**34.5.9 eMIOS CNT register n (eMIOS\_CNTn)**

This section describes the register for channels 16-23. In all other respects, this register is identical to the register for channels 0-7. See that register's description: [eMIOS CNT register n \(eMIOS\\_CNTn\)](#).

Address: 0h base + 228h offset + (32d × i), where i=0d to 7d

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   | C |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**eMIOS\_CNTn field descriptions**

| Field           | Description                                                                                              |
|-----------------|----------------------------------------------------------------------------------------------------------|
| 0–7<br>Reserved | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 8–31<br>C       | The C field contains the value of the internal counter.                                                  |

### 34.5.10 eMIOS UC Control register n (eMIOS\_Cn)

This section describes the register for channels 16-23. In all other respects, this register is identical to the register for channels 0-7. See that register's description: [eMIOS UC Control register n \(eMIOS\\_Cn\)](#).

Address: 0h base + 22Ch offset + (32d × i), where i=0d to 7d

| Bit   | 0    | 1    | 2     | 3 | 4     | 5 | 6      | 7   | 8 | 9 | 10 | 11 | 12 | 13  | 14  | 15 |
|-------|------|------|-------|---|-------|---|--------|-----|---|---|----|----|----|-----|-----|----|
| R     |      |      |       |   |       |   |        |     | 0 |   |    |    |    |     |     | 0  |
| W     | FREN | ODIS | ODISL |   | UCPRE |   | UCPREN | DMA |   |   | IF |    |    | FCK | FEN |    |
| Reset | 0    | 0    | 0     | 0 | 0     | 0 | 0      | 0   | 0 | 0 | 0  | 0  | 0  | 0   | 0   | 0  |

| Bit   | 16 | 17 | 18     | 19     | 20 | 21 | 22  | 23    | 24    | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|-------|----|----|--------|--------|----|----|-----|-------|-------|----|----|----|----|----|----|----|
| R     | 0  |    |        |        | 0  |    |     |       |       |    |    |    |    |    |    |    |
| W     |    |    | FORCMA | FORCMB |    |    | BSL | EDSEL | EDPOL |    |    |    |    |    |    |    |
| Reset | 0  | 0  | 0      | 0      | 0  | 0  | 0   | 0     | 0     | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**eMIOS\_Cn field descriptions**

| Field     | Description                                                                                                                                                                                                                                                                                         |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>FREN | Freeze Enable bit.<br><br>The FREN bit, if set and validated by FRZ bit in EMIOS_MCR register allows the channel to enter freeze state, freezing all registers values when in debug mode and allowing the MCU to perform debug functions.<br><br>0 Normal operation<br>1 Freeze UC registers values |
| 1<br>ODIS | Output Disable bit.<br><br>The ODIS bit allows disabling the output pin when running any of the output modes with the exception of GPIO mode.                                                                                                                                                       |

*Table continues on the next page...*

**eMIOS\_Cn field descriptions (continued)**

| Field         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |               |              |    |                        |    |                        |    |                        |    |                        |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|--------------|----|------------------------|----|------------------------|----|------------------------|----|------------------------|
| 0             | The output pin operates normally                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |               |              |    |                        |    |                        |    |                        |    |                        |
| 1             | If the selected Output Disable Input signal is asserted, the output pin goes to EDPOL for OPWFMB and OPWMB modes and to the complement of EDPOL for other output modes, but the Unified Channel continues to operate normally, i.e., it continues to produce FLAG and matches. When the selected Output Disable Input signal is negated, the output pin operates normally.                                                                                                                                                                                                                                                                     |               |              |    |                        |    |                        |    |                        |    |                        |
| 2–3<br>ODISSL | <p>Output Disable select bits.</p> <p>The ODISSL[ 0:1 ] bits select one of the four output disable input signals, as shown in the following table.</p> <p><b>Table 34-135. UC ODISSL selection</b></p> <table> <tr> <th>ODISSL[ 0:1 ]</th><th>input signal</th></tr> <tr> <td>00</td><td>Output Disable Input 0</td></tr> <tr> <td>01</td><td>Output Disable Input 1</td></tr> <tr> <td>10</td><td>Output Disable Input 2</td></tr> <tr> <td>11</td><td>Output Disable Input 3</td></tr> </table>                                                                                                                                              | ODISSL[ 0:1 ] | input signal | 00 | Output Disable Input 0 | 01 | Output Disable Input 1 | 10 | Output Disable Input 2 | 11 | Output Disable Input 3 |
| ODISSL[ 0:1 ] | input signal                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |               |              |    |                        |    |                        |    |                        |    |                        |
| 00            | Output Disable Input 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |               |              |    |                        |    |                        |    |                        |    |                        |
| 01            | Output Disable Input 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |               |              |    |                        |    |                        |    |                        |    |                        |
| 10            | Output Disable Input 2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |               |              |    |                        |    |                        |    |                        |    |                        |
| 11            | Output Disable Input 3                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |               |              |    |                        |    |                        |    |                        |    |                        |
| 4–5<br>UCPRE  | <p>Prescaler bits.</p> <p>The UCPRE[ 0:1 ] bits select the clock divider value for the internal prescaler of Unified Channel, as shown in the following table.</p> <p><b>Table 34-134. UC Internal Prescaler clock divider</b></p> <table> <tr> <th>UCPRE[ 0:1 ]</th><th>Divide ratio</th></tr> <tr> <td>00</td><td>1</td></tr> <tr> <td>01</td><td>2</td></tr> <tr> <td>10</td><td>3</td></tr> <tr> <td>11</td><td>4</td></tr> </table> <p><b>NOTE:</b> The UCPRE bits are mirrored in the two LSBs of the UCEXTPRE field in register EMIOS_Cn. Any write operations to UCPRE bits will affect the respective bits at EMIOS_C2n register.</p> | UCPRE[ 0:1 ]  | Divide ratio | 00 | 1                      | 01 | 2                      | 10 | 3                      | 11 | 4                      |
| UCPRE[ 0:1 ]  | Divide ratio                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |               |              |    |                        |    |                        |    |                        |    |                        |
| 00            | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |               |              |    |                        |    |                        |    |                        |    |                        |
| 01            | 2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |               |              |    |                        |    |                        |    |                        |    |                        |
| 10            | 3                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |               |              |    |                        |    |                        |    |                        |    |                        |
| 11            | 4                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |               |              |    |                        |    |                        |    |                        |    |                        |
| 6<br>UCPREN   | <p>Prescaler Enable bit.</p> <p>The UCPREN bit enables the prescaler counter.</p> <p>0 Prescaler disabled (no clock)</p> <p>1 Prescaler enabled</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |               |              |    |                        |    |                        |    |                        |    |                        |
| 7<br>DMA      | <p>Direct Memory Access bit.</p> <p>The DMA bit selects if FLAG generation will be used as an interrupt or as a DMA request.</p> <p>0 Flag/overflow assigned to Interrupt request.</p> <p>1 Flag/overflow assigned to DMA request.</p>                                                                                                                                                                                                                                                                                                                                                                                                         |               |              |    |                        |    |                        |    |                        |    |                        |
| 8<br>Reserved | <p>Reserved.</p> <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |               |              |    |                        |    |                        |    |                        |    |                        |

*Table continues on the next page...*



**eMIOS\_Cn field descriptions (continued)**

| Field                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                        |                                                  |      |                       |      |    |      |    |      |    |      |    |            |          |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|--------------------------------------------------|------|-----------------------|------|----|------|----|------|----|------|----|------------|----------|
| 9–12<br>IF             | <p>Input Filter bits.</p> <p>The IF[ 0:3 ] bits control the programmable input filter, selecting the minimum input pulse width that can pass through the filter, as shown in the following table. For output modes, these bits have no meaning.</p> <p><b>Table 34-133. UC Input Filter bits</b></p> <table> <tr> <th>IF[ 0:3 ]<sup>1</sup></th><th>Minimum input Pulse width [filter clock periods]</th></tr> <tr> <td>0000</td><td>bypassed<sup>2</sup></td></tr> <tr> <td>0001</td><td>02</td></tr> <tr> <td>0010</td><td>04</td></tr> <tr> <td>0100</td><td>08</td></tr> <tr> <td>1000</td><td>16</td></tr> <tr> <td>All others</td><td>Reserved</td></tr> </table> <p>1. Filter latency (the difference in time between the input and the response) is 3 clock edges.<br/>2. The input signal is synchronized before arriving to the digital filter.</p> | IF[ 0:3 ] <sup>1</sup> | Minimum input Pulse width [filter clock periods] | 0000 | bypassed <sup>2</sup> | 0001 | 02 | 0010 | 04 | 0100 | 08 | 1000 | 16 | All others | Reserved |
| IF[ 0:3 ] <sup>1</sup> | Minimum input Pulse width [filter clock periods]                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                        |                                                  |      |                       |      |    |      |    |      |    |      |    |            |          |
| 0000                   | bypassed <sup>2</sup>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                        |                                                  |      |                       |      |    |      |    |      |    |      |    |            |          |
| 0001                   | 02                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                        |                                                  |      |                       |      |    |      |    |      |    |      |    |            |          |
| 0010                   | 04                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                        |                                                  |      |                       |      |    |      |    |      |    |      |    |            |          |
| 0100                   | 08                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                        |                                                  |      |                       |      |    |      |    |      |    |      |    |            |          |
| 1000                   | 16                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                        |                                                  |      |                       |      |    |      |    |      |    |      |    |            |          |
| All others             | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                        |                                                  |      |                       |      |    |      |    |      |    |      |    |            |          |
| 13<br>FCK              | <p>Filter Clock select bit.</p> <p>The FCK bit selects the clock source for the programmable input filter.</p> <p>0 Prescaled clock<br/>1 eMIOS module clock</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                        |                                                  |      |                       |      |    |      |    |      |    |      |    |            |          |
| 14<br>FEN              | <p>FLAG Enable bit.</p> <p>The FEN bit allows the Unified Channel FLAG bit to generate an interrupt signal or a DMA request signal (the type of signal to be generated is defined by the DMA bit).</p> <p>0 Disable (FLAG does not generate an interrupt request)<br/>1 Enable (FLAG generates an interrupt request)</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                        |                                                  |      |                       |      |    |      |    |      |    |      |    |            |          |
| 15–17<br>Reserved      | <p>Reserved.</p> <p>This field is reserved.<br/>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                        |                                                  |      |                       |      |    |      |    |      |    |      |    |            |          |
| 18<br>FORCMA           | <p>Force Match A bit.</p> <p>For output modes, the FORCMA bit is equivalent to a successful comparison on comparator A (except that the FLAG bit is not set). This bit is cleared by reset and is always read as zero. This bit is valid for every output operation mode which uses comparator A, otherwise it has no effect.</p> <p>For input modes, the FORCMA bit is not used and writing to it has no effect.</p> <p>0 Has no effect.<br/>1 Force a match at comparator A</p>                                                                                                                                                                                                                                                                                                                                                                             |                        |                                                  |      |                       |      |    |      |    |      |    |      |    |            |          |
| 19<br>FORCMB           | <p>Force Match B bit.</p> <p>For output modes, the FORCMB bit is equivalent to a successful comparison on comparator B (except that the FLAG bit is not set). This bit is cleared by reset and is always read as zero. This bit is valid for every output operation mode which uses comparator B, otherwise it has no effect.</p> <p>0 Has not effect<br/>1 Force a match at comparator B</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                        |                                                  |      |                       |      |    |      |    |      |    |      |    |            |          |

*Table continues on the next page...*

## eMIOS\_Cn field descriptions (continued)

| Field          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |           |                           |    |                              |    |                                 |                                  |                                   |                                   |    |          |    |                                |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|---------------------------|----|------------------------------|----|---------------------------------|----------------------------------|-----------------------------------|-----------------------------------|----|----------|----|--------------------------------|
| 20<br>Reserved | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |           |                           |    |                              |    |                                 |                                  |                                   |                                   |    |          |    |                                |
| 21–22<br>BSL   | <p>Bus Select bits.</p> <p>The BSL[ 0:1 ] bits are used to select either one of the counter buses or the internal counter to be used by the Unified Channel , as shown in the following tables.</p> <p style="text-align: center;"><b>Table 34-132. UC BSL bits</b></p> <table> <tr> <th>BSL[ 0:1]</th><th>Selected bus<sup>1</sup></th></tr> <tr> <td>00</td><td>All channels: counter bus[A]</td></tr> <tr> <td rowspan="4">01</td><td>Channels 0 to 7: counter bus[B]</td></tr> <tr> <td>Channels 8 to 15: counter bus[C]</td></tr> <tr> <td>Channels 16 to 23: counter bus[D]</td></tr> <tr> <td>Channels 24 to 31: counter bus[E]</td></tr> <tr> <td>10</td><td>Reserved</td></tr> <tr> <td>11</td><td>All channels: internal counter</td></tr> </table> <p>1. Not all devices support all counter buses. See the Device Configuration chapter for details.</p> | BSL[ 0:1] | Selected bus <sup>1</sup> | 00 | All channels: counter bus[A] | 01 | Channels 0 to 7: counter bus[B] | Channels 8 to 15: counter bus[C] | Channels 16 to 23: counter bus[D] | Channels 24 to 31: counter bus[E] | 10 | Reserved | 11 | All channels: internal counter |
| BSL[ 0:1]      | Selected bus <sup>1</sup>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |           |                           |    |                              |    |                                 |                                  |                                   |                                   |    |          |    |                                |
| 00             | All channels: counter bus[A]                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |           |                           |    |                              |    |                                 |                                  |                                   |                                   |    |          |    |                                |
| 01             | Channels 0 to 7: counter bus[B]                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |           |                           |    |                              |    |                                 |                                  |                                   |                                   |    |          |    |                                |
|                | Channels 8 to 15: counter bus[C]                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |           |                           |    |                              |    |                                 |                                  |                                   |                                   |    |          |    |                                |
|                | Channels 16 to 23: counter bus[D]                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |           |                           |    |                              |    |                                 |                                  |                                   |                                   |    |          |    |                                |
|                | Channels 24 to 31: counter bus[E]                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |           |                           |    |                              |    |                                 |                                  |                                   |                                   |    |          |    |                                |
| 10             | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |           |                           |    |                              |    |                                 |                                  |                                   |                                   |    |          |    |                                |
| 11             | All channels: internal counter                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |           |                           |    |                              |    |                                 |                                  |                                   |                                   |    |          |    |                                |
| 23<br>EDSEL    | <p>Edge Selection bit.</p> <p>For input modes, the EDSEL bit selects whether the internal counter is triggered by both edges of a pulse or just by a single edge as defined by the EDPOL bit . When not shown in the mode of operation description, this bit has no effect.</p> <p>0 – Single edge triggering defined by the EDPOL bit</p> <p>1 – Both edges triggering{mts_ipi_slv_saic}</p> <p>For GPIO in mode, the EDSEL bit selects if a FLAG can be generated.</p> <p>0 – A FLAG is generated as defined by the EDPOL bit</p> <p>1 – No FLAG is generated</p> <p>For SAOC mode, the EDSEL bit selects the behavior of the output flip-flop at each match.</p> <p>0 – The EDPOL value is transferred to the output flip-flop</p> <p>1 – The output flip-flop is toggled</p>                                                                                     |           |                           |    |                              |    |                                 |                                  |                                   |                                   |    |          |    |                                |
| 24<br>EDPOL    | <p>Edge Polarity bit.</p> <p>For input modes (except QDEC mode), the EDPOL bit asserts which edge triggers either the internal counter or an input capture or a FLAG . When not shown in the mode of operation description, this bit has no effect.</p> <p>0 – Trigger on a falling edge</p> <p>1 – Trigger on a rising edge</p> <p>For QDEC(MODE[0] cleared), the EDPOL bit selects the count direction according to <i>direction</i> signal (UC[n] input).</p> <p>1 - counts up when UC[n] is asserted</p> <p>0 - counts down when UC[n] is asserted</p>                                                                                                                                                                                                                                                                                                           |           |                           |    |                              |    |                                 |                                  |                                   |                                   |    |          |    |                                |

Table continues on the next page...

**eMIOS\_Cn field descriptions (continued)**

| Field                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|--------------------------------|---------|-------------------------------------------|---------|--------------------------------------------|---------|-----------------------------|---------|------------------------------|---------|-------------------------------|---------|--------------------------|---------|---------------------------------------------------------|---------|------------------------------------------------------------|---------|--------------------------------------|---------|---------------------------------------|---------|----------------------------------|---------|-----------------------------------|---------|-----------------------------------------------------------|---------|-----------------------------------------------------------|---------|-----------------------------------------|---------|----------|----------------------|--------------------------------------------------------|---------|------------------------------------------------------|----------------------|-----------------------------------|----------------------|----------------------------------------------------------------|----------------------|------------------------------------------------------------------|
|                      | <p><b>NOTE:</b> UC[n-1] EDPOL bit selects which edge clocks the internal counter of UC[n]</p> <p>For QDEC(MODE[0] cleared), the EDPOL bit selects the count direction according to the phase difference.</p> <p>1 - Internal counter increments if phase_A is ahead phase_B signal</p> <p>0 - Internal counter decrements if phase_A is ahead phase_B signal</p> <p><b>NOTE:</b> In order to operate properly, EDPOL bit must contain the same value in UC[n] and UC[n-1]</p> <p>For output modes, the EDPOL bit is used to select the logic level on the output pin .</p> <p>0 – A match on comparator A clears the output flip-flop, while a match on comparator B sets it</p> <p>1 – A match on comparator A sets the output flip-flop, while a match on comparator B clears it</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |
| 25–31<br>MODE        | <p>Mode selection bits.</p> <p>The MODE bits select the mode of operation of the Unified Channel, as shown in the following table. Values not listed are reserved.</p> <p><b>NOTE:</b> If a reserved value is written to the mode bit field, the results are unpredictable.</p> <p style="text-align: center;"><b>Table 34-131. UC MODE bits</b></p> <table> <tr> <th>MODE[ 0:6]</th><th>Mode of operation<sup>1</sup></th></tr> <tr><td>0000000</td><td>General purpose Input/Output mode (input)</td></tr> <tr><td>0000001</td><td>General purpose Input/Output mode (output)</td></tr> <tr><td>0000010</td><td>Single Action Input Capture</td></tr> <tr><td>0000011</td><td>Single Action Output Compare</td></tr> <tr><td>0000100</td><td>Input Pulse Width Measurement</td></tr> <tr><td>0000101</td><td>Input Period Measurement</td></tr> <tr><td>0000110</td><td>Double Action Output compare (with FLAG set on B match)</td></tr> <tr><td>0000111</td><td>Double Action Output compare (with FLAG set on both match)</td></tr> <tr><td>0001000</td><td>Pulse/Edge Accumulation (continuous)</td></tr> <tr><td>0001001</td><td>Pulse/Edge Accumulation (single shot)</td></tr> <tr><td>0001010</td><td>Pulse/Edge Counting (continuous)</td></tr> <tr><td>0001011</td><td>Pulse/Edge Counting (single shot)</td></tr> <tr><td>0001100</td><td>Quadrature Decode (for count and direction encoders type)</td></tr> <tr><td>0001101</td><td>Quadrature Decode (for phase_A and phase_B encoders type)</td></tr> <tr><td>0001110</td><td>Windowed Programmable Time Accumulation</td></tr> <tr><td>0001111</td><td>Reserved</td></tr> <tr><td>001000b<sup>2</sup></td><td>Modulus Counter (Up counter with clear on match start)</td></tr> <tr><td>001001b</td><td>Modulus Counter (Up counter with clear on match end)</td></tr> <tr><td>00101bb<sup>2</sup></td><td>Modulus Counter (Up/Down counter)</td></tr> <tr><td>00110b0<sup>2</sup></td><td>Output Pulse Width and Frequency Modulation (immediate update)</td></tr> <tr><td>00110b1<sup>2</sup></td><td>Output Pulse Width and Frequency Modulation (next period update)</td></tr> </table> | MODE[ 0:6] | Mode of operation <sup>1</sup> | 0000000 | General purpose Input/Output mode (input) | 0000001 | General purpose Input/Output mode (output) | 0000010 | Single Action Input Capture | 0000011 | Single Action Output Compare | 0000100 | Input Pulse Width Measurement | 0000101 | Input Period Measurement | 0000110 | Double Action Output compare (with FLAG set on B match) | 0000111 | Double Action Output compare (with FLAG set on both match) | 0001000 | Pulse/Edge Accumulation (continuous) | 0001001 | Pulse/Edge Accumulation (single shot) | 0001010 | Pulse/Edge Counting (continuous) | 0001011 | Pulse/Edge Counting (single shot) | 0001100 | Quadrature Decode (for count and direction encoders type) | 0001101 | Quadrature Decode (for phase_A and phase_B encoders type) | 0001110 | Windowed Programmable Time Accumulation | 0001111 | Reserved | 001000b <sup>2</sup> | Modulus Counter (Up counter with clear on match start) | 001001b | Modulus Counter (Up counter with clear on match end) | 00101bb <sup>2</sup> | Modulus Counter (Up/Down counter) | 00110b0 <sup>2</sup> | Output Pulse Width and Frequency Modulation (immediate update) | 00110b1 <sup>2</sup> | Output Pulse Width and Frequency Modulation (next period update) |
| MODE[ 0:6]           | Mode of operation <sup>1</sup>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |
| 0000000              | General purpose Input/Output mode (input)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |
| 0000001              | General purpose Input/Output mode (output)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |
| 0000010              | Single Action Input Capture                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |
| 0000011              | Single Action Output Compare                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |
| 0000100              | Input Pulse Width Measurement                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |
| 0000101              | Input Period Measurement                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |
| 0000110              | Double Action Output compare (with FLAG set on B match)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |
| 0000111              | Double Action Output compare (with FLAG set on both match)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |
| 0001000              | Pulse/Edge Accumulation (continuous)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |
| 0001001              | Pulse/Edge Accumulation (single shot)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |
| 0001010              | Pulse/Edge Counting (continuous)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |
| 0001011              | Pulse/Edge Counting (single shot)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |
| 0001100              | Quadrature Decode (for count and direction encoders type)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |
| 0001101              | Quadrature Decode (for phase_A and phase_B encoders type)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |
| 0001110              | Windowed Programmable Time Accumulation                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |
| 0001111              | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |
| 001000b <sup>2</sup> | Modulus Counter (Up counter with clear on match start)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |
| 001001b              | Modulus Counter (Up counter with clear on match end)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |
| 00101bb <sup>2</sup> | Modulus Counter (Up/Down counter)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |
| 00110b0 <sup>2</sup> | Output Pulse Width and Frequency Modulation (immediate update)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |
| 00110b1 <sup>2</sup> | Output Pulse Width and Frequency Modulation (next period update)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |            |                                |         |                                           |         |                                            |         |                             |         |                              |         |                               |         |                          |         |                                                         |         |                                                            |         |                                      |         |                                       |         |                                  |         |                                   |         |                                                           |         |                                                           |         |                                         |         |          |                      |                                                        |         |                                                      |                      |                                   |                      |                                                                |                      |                                                                  |

Table continues on the next page...

**eMIOS\_Cn field descriptions (continued)**

| Field | Description                                                                                                                                                                                              |                                                                                   |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|
|       | <b>Table 34-131. UC MODE bits (continued)</b>                                                                                                                                                            |                                                                                   |
|       | <b>MODE[ 0:6]</b>                                                                                                                                                                                        | <b>Mode of operation<sup>1</sup></b>                                              |
|       | 00111b0 <sup>2</sup>                                                                                                                                                                                     | Center Aligned Output Pulse Width Modulation (with trail edge dead-time)          |
|       | 00111b1 <sup>2</sup>                                                                                                                                                                                     | Center Aligned Output Pulse Width Modulation (with lead edge dead-time)           |
|       | 01000b0 <sup>2</sup>                                                                                                                                                                                     | Output Pulse Width Modulation (immediate update)                                  |
|       | 01000b1 <sup>2</sup>                                                                                                                                                                                     | Output Pulse Width Modulation (next period update)                                |
|       | 0100100 through 0100101                                                                                                                                                                                  | Reserved                                                                          |
|       | 0100110                                                                                                                                                                                                  | Output Pulse Width Modulation with Trigger                                        |
|       | 0100111 through 1001111                                                                                                                                                                                  | Reserved                                                                          |
|       | 101000b <sup>2</sup>                                                                                                                                                                                     | Modulus Counter Buffered (Up counter)                                             |
|       | 101001b <sup>2</sup>                                                                                                                                                                                     | Reserved                                                                          |
|       | 10101bb <sup>2</sup>                                                                                                                                                                                     | Modulus Counter Buffered (Up/Down counter)                                        |
|       | 10110b0 <sup>2</sup>                                                                                                                                                                                     | Output Pulse Width and Frequency Modulation Buffered                              |
|       | 10110b1 <sup>2</sup>                                                                                                                                                                                     | Reserved                                                                          |
|       | 10111b0 <sup>2</sup>                                                                                                                                                                                     | Center Aligned Output Pulse Width Modulation Buffered (with trail edge dead-time) |
|       | 10111b1 <sup>2</sup>                                                                                                                                                                                     | Center Aligned Output Pulse Width Modulation Buffered (with lead edge dead-time)  |
|       | 11000b0 <sup>2</sup>                                                                                                                                                                                     | Output Pulse Width Modulation Buffered                                            |
|       | 1100001 through 1111111                                                                                                                                                                                  | Reserved                                                                          |
|       | 1. Not all modes of operation are supported. See the configuration information for details.<br>2. b = adjust parameters for the mode of operation. Refer to “UC Modes of Operation” section for details. |                                                                                   |

1. Filter latency (the difference in time between the input and the response) is 3 clock edges.
2. The input signal is synchronized before arriving to the digital filter.
3. Not all devices support all counter buses. See the Device Configuration chapter for details.
4. Not all modes of operation are supported. See the configuration information for details.
5. b = adjust parameters for the mode of operation. Refer to “UC Modes of Operation” section for details.

**34.5.11 eMIOS UC Status register n (eMIOS\_Sn)**

This section describes the register for channels 16-23. For channels 0-7, see [eMIOS UC Status register n \(eMIOS\\_Sn\)](#).

The UC Status register EMIOS\_Sn gathers bits reflecting the status of the UC input/output signals and the overflow condition of the internal counter.

Address: 0h base + 230h offset + (32d × i), where i=0d to 7d

|       |      |    |    |    |    |    |    |    |    |    |    |    |    |      |       |      |
|-------|------|----|----|----|----|----|----|----|----|----|----|----|----|------|-------|------|
| Bit   | 0    | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13   | 14    | 15   |
| R     | OVR  | 0  |    |    |    |    |    |    |    |    |    |    |    |      |       |      |
| W     | w1c  |    |    |    |    |    |    |    |    |    |    |    |    |      |       |      |
| Reset | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0     | 0    |
| Bit   | 16   | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29   | 30    | 31   |
| R     | OVFL | 0  |    |    |    |    |    |    |    |    |    |    |    | UCIN | UCOUT | FLAG |
| W     | w1c  |    |    |    |    |    |    |    |    |    |    |    |    |      |       | w1c  |
| Reset | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | *    | 0     | 0    |

\* Notes:

- UCIN field: Reset value is undefined.

### eMIOS\_*Sn* field descriptions

| Field             | Description                                                                                                                                                                                                                                                                                   |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>OVR          | <p>Overflow bit.</p> <p>The OVR bit indicates that FLAG generation occurred when the FLAG bit was already set .</p> <p>The OVR bit can be cleared either by clearing the FLAG bit or by writing a 1 to the OVR bit.</p> <p>0   Overflow has not occurred</p> <p>1   Overflow has occurred</p> |
| 1–15<br>Reserved  | <p>Reserved.</p> <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                           |
| 16<br>OVFL        | <p>Overflow bit.</p> <p>The OVFL bit indicates that an overflow has occurred in the internal counter.</p> <p>The OVFL bit must be cleared by software writing a 1 to it.</p> <p>0   No overflow</p> <p>1   An overflow had occurred</p>                                                       |
| 17–28<br>Reserved | <p>Reserved.</p> <p>This field is reserved.</p>                                                                                                                                                                                                                                               |

*Table continues on the next page...*

**eMIOS\_Sn field descriptions (continued)**

| Field       | Description                                                                                                                                                                                                       |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|             | This read-only field is reserved and always has the value 0.                                                                                                                                                      |
| 29<br>UCIN  | Unified Channel Input pin bit.<br>The UCIN bit reflects the input pin state after being filtered and synchronized.                                                                                                |
| 30<br>UCOUT | Unified Channel Output pin bit.<br>The UCOUT bit reflects the output pin state .                                                                                                                                  |
| 31<br>FLAG  | FLAG bit.<br>The FLAG bit is set when an input capture or a match event in the comparators occurred.<br>The FLAG bit must be cleared by writing a 1 to it.<br><br>0 FLAG cleared<br>1 FLAG set event has occurred |

**34.5.12 eMIOS Alternate Address register n (eMIOS\_ALTAn)**

This section describes the register for channels 16-23. In all other respects, this register is identical to the register for channels 0-7. See that register's description: [eMIOS Alternate Address register n \(eMIOS\\_ALTAn\)](#).

Address: 0h base + 234h offset + (32d × i), where i=0d to 7d

|       |   |   |   |   |   |   |   |   |      |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|------|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8    | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   | ALTA |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |      |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0    | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**eMIOS\_ALTAn field descriptions**

| Field           | Description                                                                                          |
|-----------------|------------------------------------------------------------------------------------------------------|
| 0-7<br>Reserved | Reserved.<br>This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 8-31<br>ALTA    | See text above register figure.                                                                      |

**34.6 Functional Description**

The eMIOS provides independent channels (Unified) that can be configured and accessed by a host MCU. Up to five time bases can be shared by the channels.

The eMIOS module is based on a multi-bus timer architecture in which several timer channels are used to drive counter buses that are shared among the channels. Up to five timebases can be shared by the channels through five counter buses (counter bus A or F, B, C, D and E) and each channel can generate its own timebase. There are five counter buses in the module: one global counter bus, shared by all channels and selectable among time bases A or F, and four local counter buses (B, C, D, and E), each one dedicated to a slice of 8 channels. Counter bus A is referred to as the global counter bus. Counter buses B, C, D, and E are the local counter buses.

The eMIOS counter buses are driven by channels in specific locations. Global counter bus A is driven by the channel in channel slot [23].. Counter buses B, C, D, and E are driven by channels in slots 0, 8, 16, and 24, respectively. Counter bus A or F drives all channels. Counter bus B drives channels in slots from 0 through 7. Counter bus C drives channels in slots from 8 through 15. Counter bus D drives channels in slots from 16 through 23. Counter bus E drives channels from 24 through 31.

The eMIOS block is reset asynchronously. All registers are cleared on reset.

The following figure shows an eMIOS block configured with a mix of Unified Channels. In this configuration example, each one of the channels can be driven by the respective local bus—B for the [7:0] slice and D for the [23:16] slice—or by the global counter bus A. Note that there are Unified Channels with no external pin associated. These channels are used to trigger internal SoC signals such as interrupt requests or DMA transfer requests. Since the channel flags are also connected to the block top-level pins, they can be used as trigger signals for other blocks at chip level.

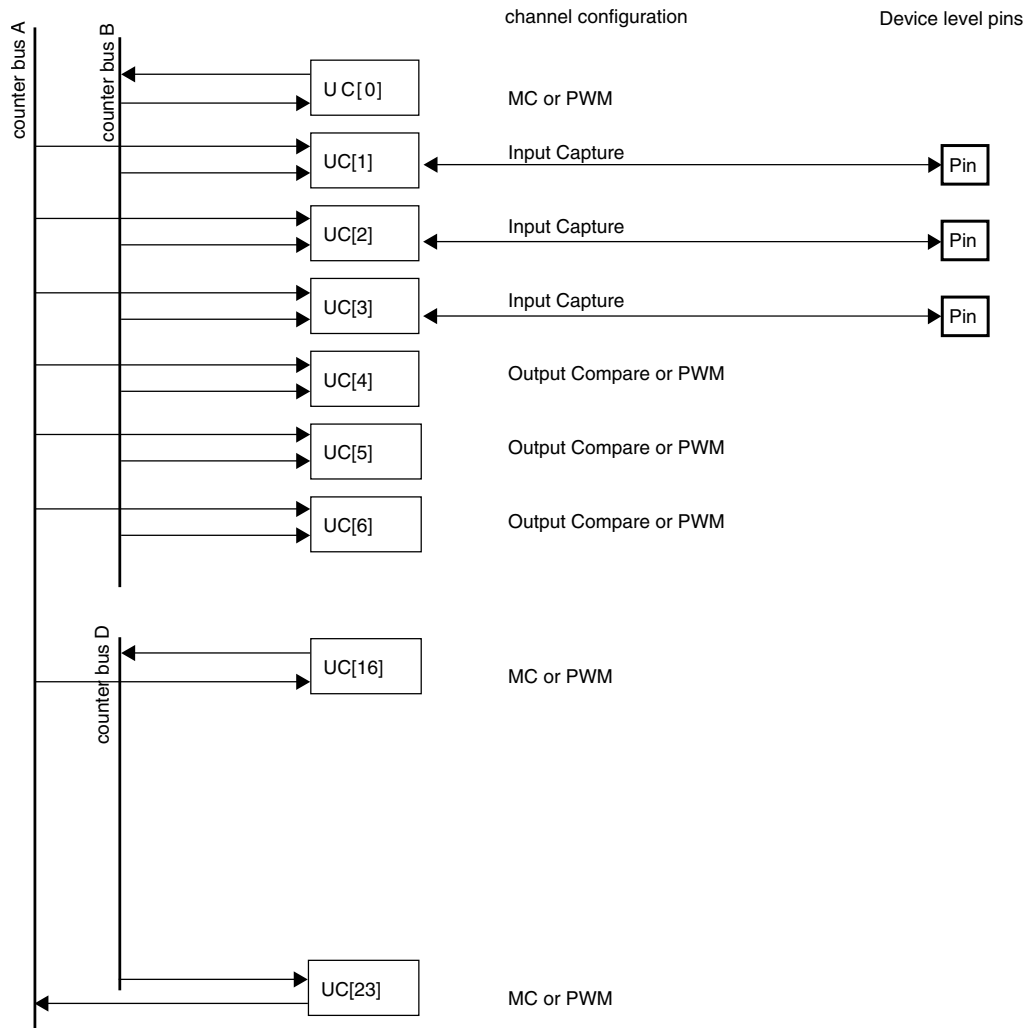


**Figure 34-113. eMIOS example configuration using Unified Channels**

## 34.6.1 Unified Channel (UC)

Figure 34-114 shows the Unified Channel block diagram. Each Unified Channel consists of the following:

- A counter bus selector, which selects the timebase to be used by the channel for all timing functions
- A programmable clock prescaler
- Two double-buffered data registers, A and B, that allow up to two input capture and/or output compare events to occur before software intervention is needed
- Two comparators (equal only), A and B, that compare the selected counter bus with the value in the data registers



- An internal counter, which can be used as a local time base or to count input events
- Programmable input filter, which ensures that only valid pin transitions are received by channel
- Programmable input edge detector, which detects the rising, falling, or either edge
- An output flip-flop, which holds the logic level to be applied to the output pin
- eMIOS Status and Control register
- An Output Disable Input selector, which selects the Output Disable Input signal that will be used as output disable

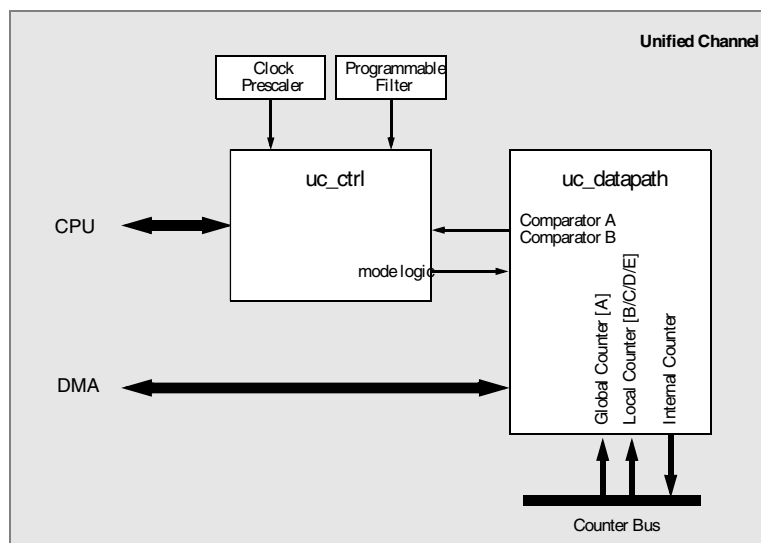


**Figure 34-114. Unified Channel block diagram**

Figure 34-115 shows both the Unified Channel control and datapath block diagrams. The control sub-block is responsible for the generation of signals to control the multiplexers in the datapath sub-block. Each mode is implemented by a dedicated logic independent from others modes, thus allowing to optimize the logic by disabling the mode and therefore its associated logic. Targeting the logic optimization, a set of registers is shared by the modes thus providing sequential events to be stored.

The datapath block provides the channel A and B registers, the internal timebase and comparators. Multiplexers select the input of comparators and data for the registers inputs, thus configuring the datapath to implement the channel modes. The outputs of the A and B comparators are connected to the uc\_ctrl control block.

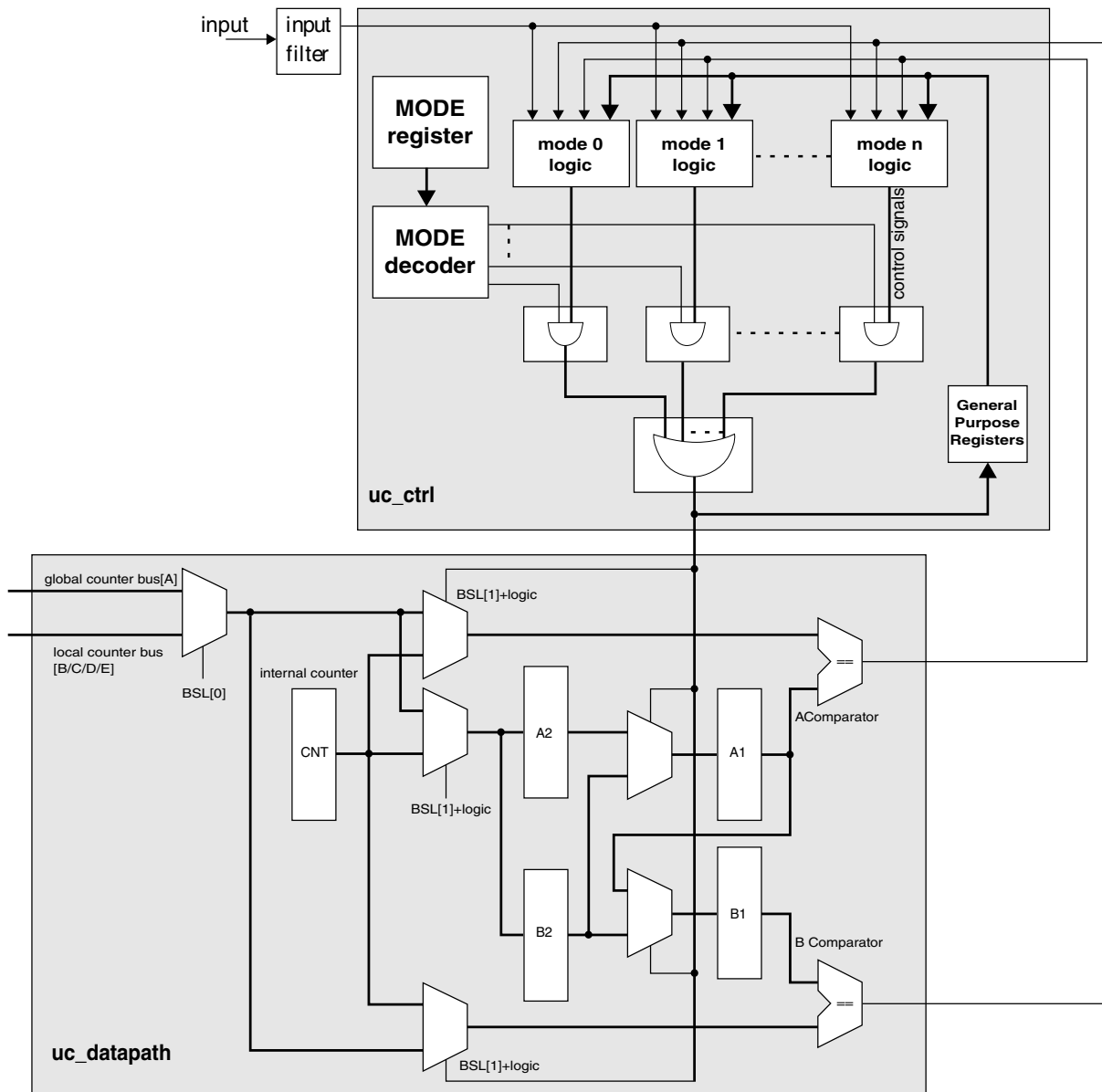


Figure 34-115. Unified Channel control and datapath block diagrams

### 34.6.1.1 Unified Channel modes of operation

The mode of operation of the Unified Channel is determined by the mode select bits `MODE[0:6]` in the `EMIOS_C_n` register.

When entering an output mode (except for GPIO mode), the output flip-flop is set to the disabled state according to `EMIOSCn[ODIS]`.

As the internal counter, `EMIOSCNTn`, runs in all modes (except for GPIO mode) and can be used as a timebase if the resource is not used in the current mode.

In order to provide smooth waveform generation even when A and B registers are immediately changed, it is available with the MCB, OPWFMB, OPWMB and OPWFMB modes. In these modes, A and B registers are double buffered. They are presented in separate sections because there are several basic differences between these and each corresponding nonbuffered mode.

#### 34.6.1.1.1 General-Purpose Input/Output (GPIO) mode

In GPIO mode, all input capture and output compare functions of the Unified Channel are disabled, and the internal counter (EMIOSCNTn register) is cleared and disabled. All control bits remain accessible. In order to prepare the UC for a new operation mode, writing to registers EMIOSAn or EMIOSBn stores the same value in registers A1/A2 or B1/B2, respectively. Writing to the EMIOSALTAn register stores a value only in register A2.

MODE[6] bit selects between input (MODE[6] = 0) and output (MODE[6] = 1) modes.

When changing MODE[0:6], the application software must first enter GPIO mode to reset the Unified Channel's internal functions properly. Failure to do this could lead to invalid and unexpected output compare or input capture results or the FLAGs being set incorrectly.

In GPIO input mode (MODE[0:6] = 0000000), FLAG generation is determined according to EDPOL and EDSEL bits and the input pin status can be determined by reading the UCIN bit.

In GPIO output mode (MODE[0:6] = 0000001), the Unified Channel is used as a single output port pin and the value of the EDPOL bit is permanently transferred to the output flip-flop.

#### 34.6.1.1.2 Single Action Input Capture (SAIC) Mode

In SAIC mode (MODE[0:6]=0000010), when a triggering event occurs on the input pin, the value on the selected time base is captured into register A2. The FLAG bit is set along with the capture event to indicate that an input capture has occurred. Register EMIOSA[n] returns the value of register A2. As soon as the SAIC mode is entered coming out from GPIO mode the channel is ready to capture events. The events are captured as soon as they occur thus reading register A always returns the value of the latest captured event. Subsequent captures are enabled with no need of further reads from EMIOSA[n] register. The FLAG is set at any time a new event is captured.

The input capture is triggered by a rising, falling or either edges in the input pin, as configured by EDPOL and EDSEL bits in EMIOSC[n] register.

Figure 34-116 and Figure 34-117 show how the Unified Channel can be used for input capture.

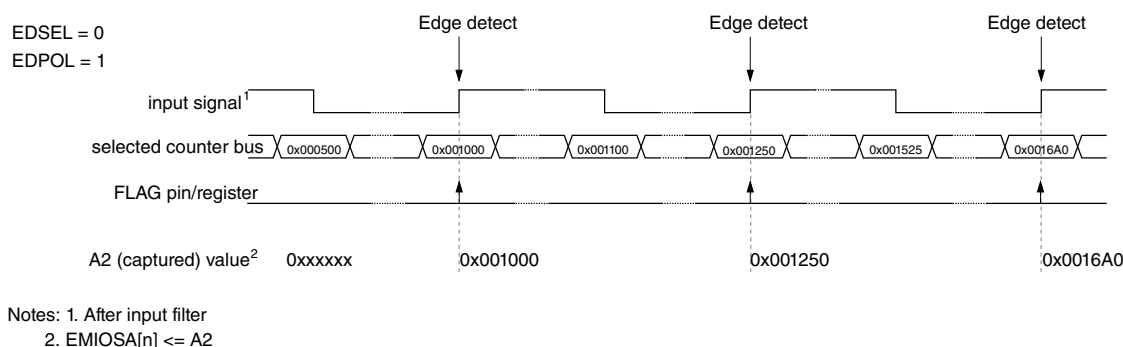


**Figure 34-116. Single Action Input Capture with rising edge triggering example**

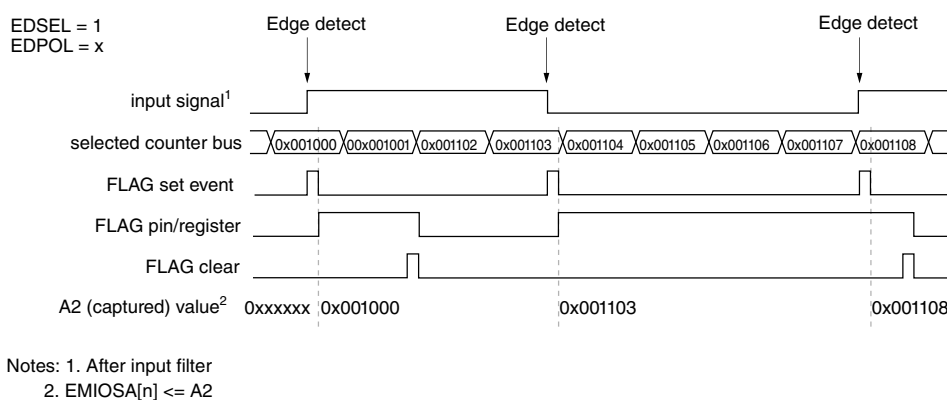


**Figure 34-117. Single Action Input Capture with both edges triggering example**

### 34.6.1.1.3 Single Action Output Compare (SAOC) Mode

In SAOC mode (MODE[0:6]=0000011) a match value is loaded in register A2 and then immediately transferred to register A1 to be compared with the selected time base. When a match occurs, the EDSEL bit selects whether the output flip-flop is toggled or the value in EDPOL is transferred to it. Along with the match the FLAG bit is set to indicate that the output compare match has occurred. Writing to register EMIOSA[n] stores the value in register A2 and reading to register EMIOSA[n] returns the value of register A1.

An output compare match can be simulated in software by setting the FORCMA bit in EMIOSC[n] register. In this case, the FLAG bit is not set.

When SAOC mode is entered coming out from GPIO mode, the output flip-flop is set to the complement of the EDPOL bit in the EMIOSC[n] register.

Counter bus can be either internal or external and is selected through BSL[0:1] bits.

Figure 34-118 and Figure 34-119 show how the Unified Channel can be used to perform a single output compare with EDPOL value being transferred to the output flip-flop and toggling the output flip-flop at each match, respectively. Note that once in SAOC mode the matches are enabled thus the desired match value on register A1 must be written before the mode is entered. A1 register can be updated at any time thus modifying the match value which will reflect in the output signal generated by the channel. Subsequent matches are enabled with no need of further writes to EMIOSA[n] register. The FLAG is set at the same time a match occurs (see Figure 34-120).

### Note

The channel internal counter in SAOC mode is free-running. It starts counting as soon as the SAOC mode is entered.

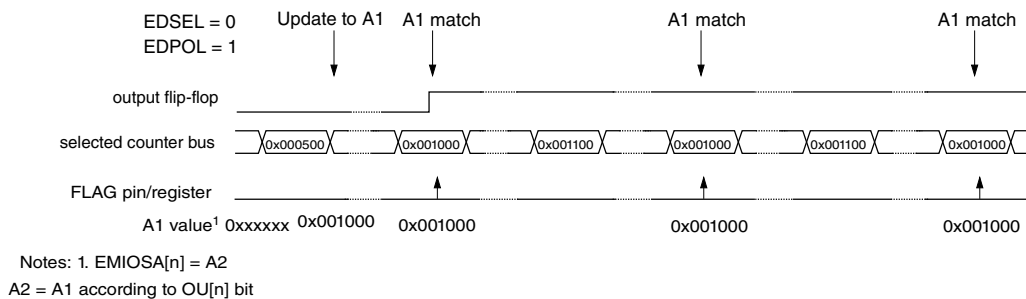


**Figure 34-118. SAOC example with EDPOL value being transferred to the output flip-flop**

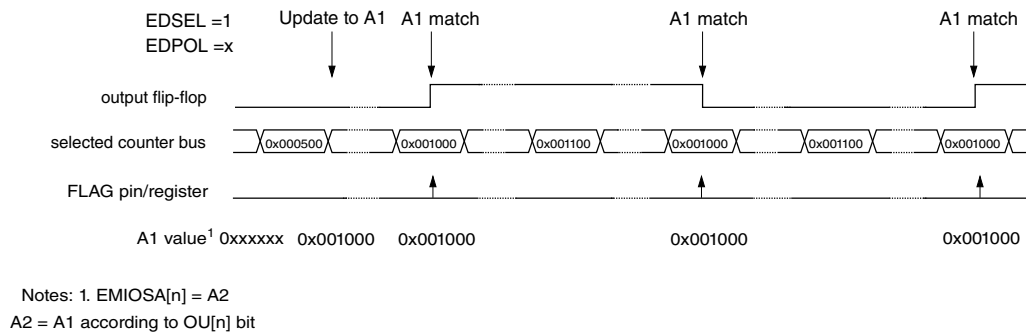


**Figure 34-119. SAOC example toggling the output flip-flop**

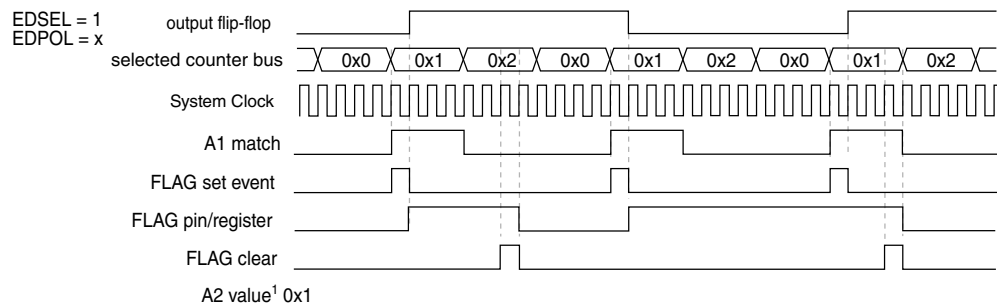


**Figure 34-120. SAOC example with flag behavior**

#### 34.6.1.1.4 Input Pulse Width Measurement (IPWM) Mode

The IPWM mode (MODE[0:6]=0000100) allows the measurement of the width of a positive or negative pulse by capturing the leading edge on register B1 and the trailing edge on register A2. Successive captures are done on consecutive edges of opposite polarity. The leading edge sensitivity (i.e., pulse polarity) is selected by EDPOL bit in the EMIOSC[n] register. Registers EMIOSA[n] and EMIOSB[n] return the values in register A2 and B1, respectively.

The capture function of register A2 remains disabled until the first leading edge triggers the first input capture on register B2. When this leading edge is detected, the count value of the selected time base is latched into register B2; the FLAG bit is not set. When the trailing edge is detected, the count value of the selected time base is latched into register A2 and, at the same time, the FLAG bit is set and the content of register B2 is transferred to register B1 and to register A1.

If subsequent input capture events occur while the corresponding FLAG bit is set, registers A2, B1 and A1 will be updated with the latest captured values and the FLAG will remain set. Registers EMIOSA[n] and EMIOSB[n] return the value in registers A2 and B1, respectively.

In order to guarantee coherent access, reading EMIOSA[n] forces B1 be updated with the content of register A1. At the same time transfers between B2 and B1 are disabled until the next read of EMIOSB[n] register. Reading EMIOSB[n] register forces B1 be updated with A1 register content and re-enables transfers from B2 to B1, to take effect at the next trailing edge capture. Transfers from B2 to A1 are not blocked at any time.

The input pulse width is calculated by subtracting the value in B1 from A2.

The following figure shows how the Unified Channel can be used for input pulse width measurement.

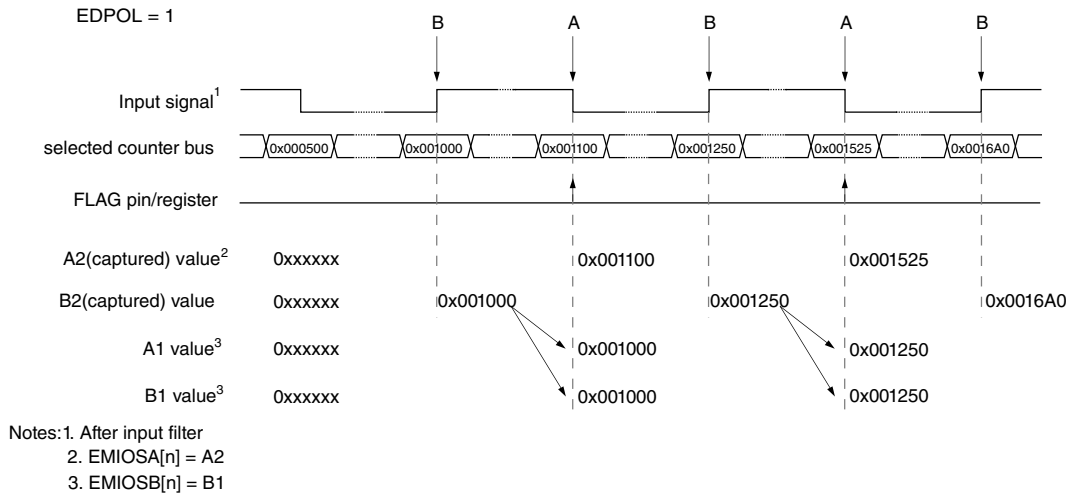


**Figure 34-121. Input Pulse Width Measurement example**

The following figure shows the A1 and B1 updates when EMIOSA[n] and EMIOB[n] register reads occur. Note that A1 register has always coherent data related to A2 register. Note also that when EMIOSA[n] read is performed B1 register is loaded with A1 register content. This guarantee that the data in register B1 has always the coherent data related to the last EMIOSA[n] read. The B1 register updates remains locked until EMIOB[n] read occurs. If EMIOSA[n] read is performed B1 is updated with A1 register content even if B1 update is locked by a previous EMIOSA[n] read operation.

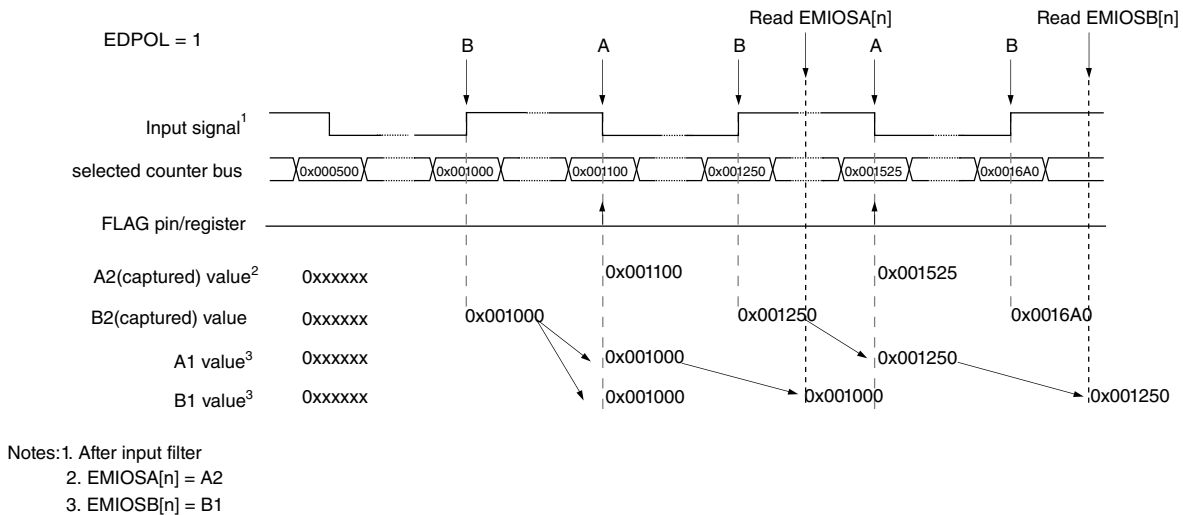


**Figure 34-122. B1 and A1 updates at EMIOSA[n] and EMIOB[n] reads**

Reading EMIOS\_A\_n followed by EMIOS\_B\_n always provides coherent data. If coherent data is not required for any reason, the sequence of reads should be inverted, therefore EMIOB[n] should be read prior to EMIOSA[n] register. Note that even in this case B1 register updates will be blocked after EMIOSA[n] read, thus a second EMIOB[n] is required in order to release B1 register updates.

### 34.6.1.1.5 Input Period Measurement (IPM) Mode

The IPM mode (MODE[0:6]=0000101) allows the measurement of the period of an input signal by capturing two consecutive rising edges or two consecutive falling edges. Successive input captures are done on consecutive edges of the same polarity. The edge polarity is defined by the EDPOL bit in the EMIOSC[n] register.

#### Note

The input signal must have at least four system clock cycles period in order to be properly captured by the synchronization logic at the channel input even if the input filter is in by-pass mode.

When the first edge of selected polarity is detected, the selected time base is latched into the registers A2 and B2, and the data previously held in register B2 is transferred to register B1. On this first capture the FLAG line is not set, and the values in registers B1 is meaningless. On the second and subsequent captures, the FLAG line is set and data in register B2 is transferred to register B1.

When the second edge of the same polarity is detected, the counter bus value is latched into registers A2 and B2, the data previously held in register B2 is transferred to data register B1 and to register A1. The FLAG bit is set to indicate the start and end points of a complete period have been captured. This sequence of events is repeated for each subsequent capture. Registers EMIOSA[n] and EMIOSB[n] return the values in register A2 and B1, respectively.

In order to allow coherent data, reading EMIOSA[n] forces A1 content be transferred to B1 register and disables transfers between B2 and B1. These transfers are disabled until the next read of the EMIOSB[n] register. Reading EMIOSB[n] register forces A1 content to be transferred to B1 and re-enables transfers from B2 to B1, to take effect at the next edge capture.

The input pulse period is calculated by subtracting the value in B1 from A2.

The following figure shows how the Unified Channel can be used for input period measurement.



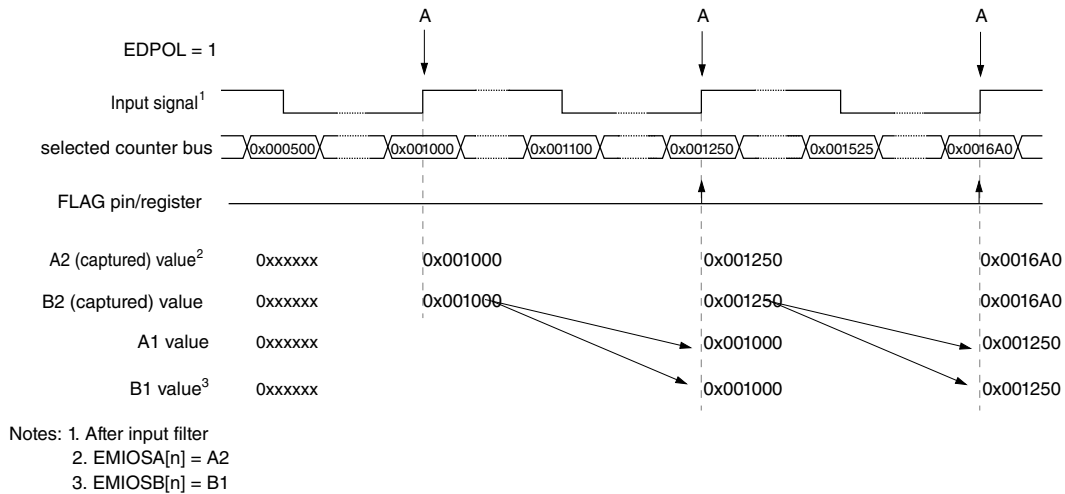


Figure 34-123. Input Period Measurement example

Figure 34-124 describes the A1 and B1 register updates when EMIOSA[n] and EMIOSB[n] read operations are performed. When EMIOSA[n] read occurs the content of A1 is transferred to B1 thus providing coherent data in A2 and B1 registers. Transfers from B2 to B1 are then blocked until EMIOSB[n] is read. After EMIOSB[n] is read, register A1 content is transferred to register B1 and the transfers from B2 to B1 are re-enabled to occur at the transfer edges, which is the leading edge in the following example figure.

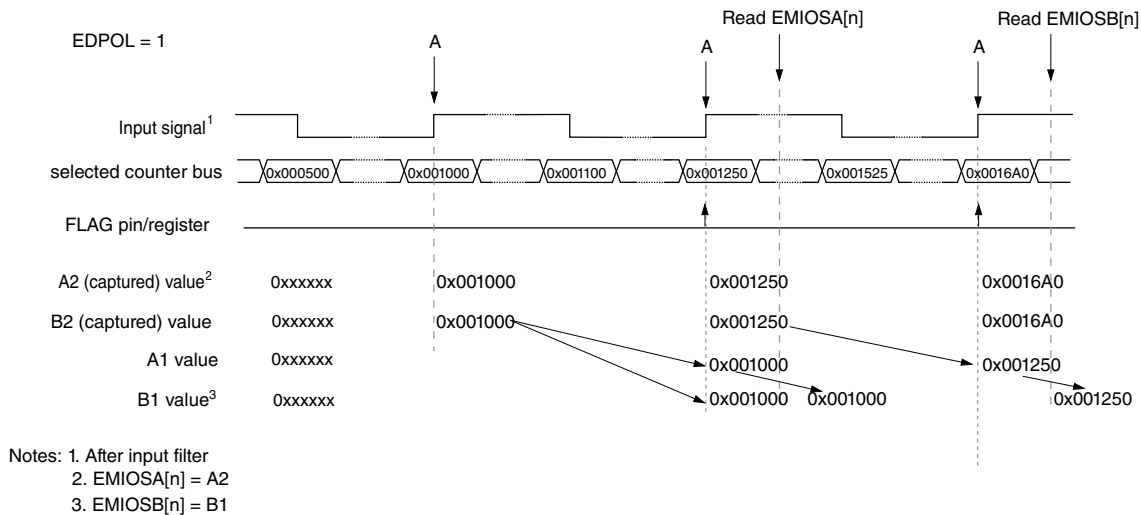


Figure 34-124. A1 and B1 updates at EMIOSA[n] and EMIOSB[n] reads

#### 34.6.1.1.6 Double Action Output Compare (DAOC) Mode

In the DAOC mode, the leading and trailing edges of the variable pulse width output are generated by matches occurring on comparators A and B. There is no restriction concerning the order in which A and B matches occur.

When the DAOC mode is entered, coming out from GPIO mode both comparators are disabled and the output flip-flop is set to the complement of the EDPOL bit in the EMIOSC[n] register.

Data written to A2 and B2 are transferred to A1 and B1, respectively, on the next system clock cycle if OU[n] bit of EMIOSOUDIS register is cleared (see ). The transfer is blocked if OU[n] bit is set. Comparator A is enabled only after the transfer to A1 register occurs and is disabled on the next A match. Comparator B is enabled only after the transfer to B1 register occurs and is disabled on the next B match. Comparators A and B are enabled and disabled independently.

The output flip-flop is set to the value of EDPOL when a match occurs on comparator A and to the complement of EDPOL when a match occurs on comparator B.

MODE[6] controls if the FLAG is set on both matches (MODE[0:6]=0000111) or just on the B match (MODE[0:6]=0000110). FLAG bit assertion depends on comparator enabling.

If subsequent enabled output compares occur on registers A1 and B1, pulses will continue to be generated, regardless of the state of the FLAG bit.

At any time, the FORCMA and FORCMB bits allow the software to force the output flip-flop to the level corresponding to a comparison event in comparator A or B, respectively. Note that the FLAG bit is not affected by these forced operations.

### **Note**

If both registers (A1 and B1) are loaded with the same value, the B match prevails concerning the output pin state (output flip-flop is set to the complement of EDPOL), the FLAG bit is set and both comparators are disabled.

The following two tables show how the Unified Channel can be used to generate a single output pulse with FLAG bit being set on the second match or on both matches, respectively.

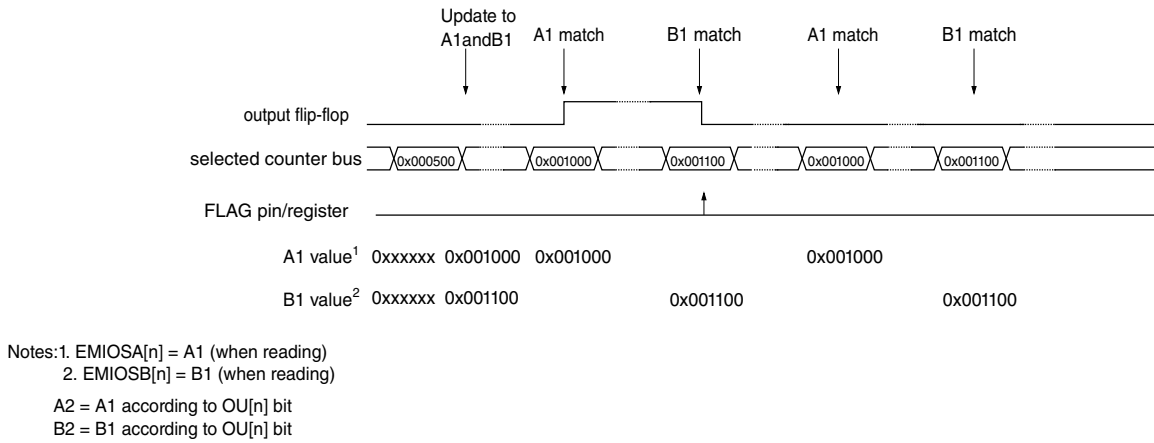


**Figure 34-125. Double Action Output Compare with FLAG set on the second match, MODE[0] bit is 0**

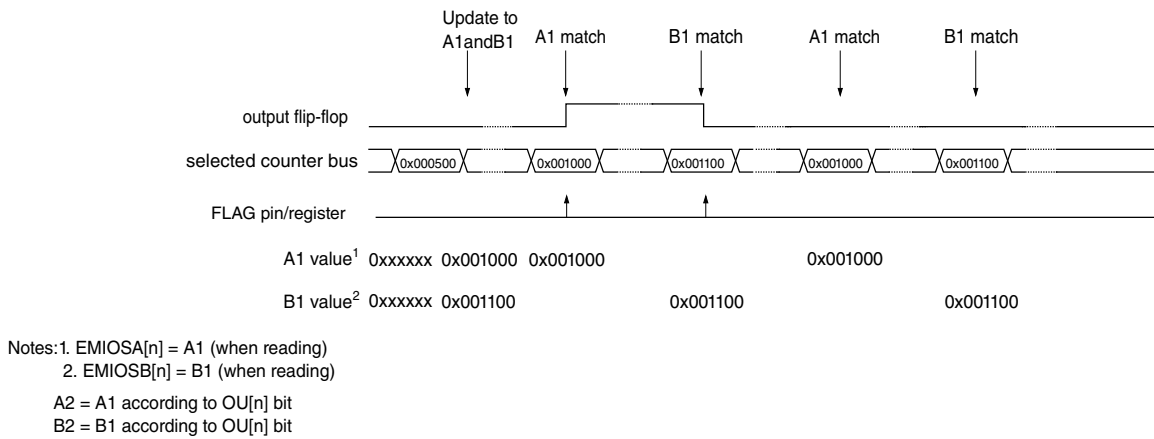


**Figure 34-126. Double Action Output Compare with FLAG set on both matches, MODE[0] bit is 1**

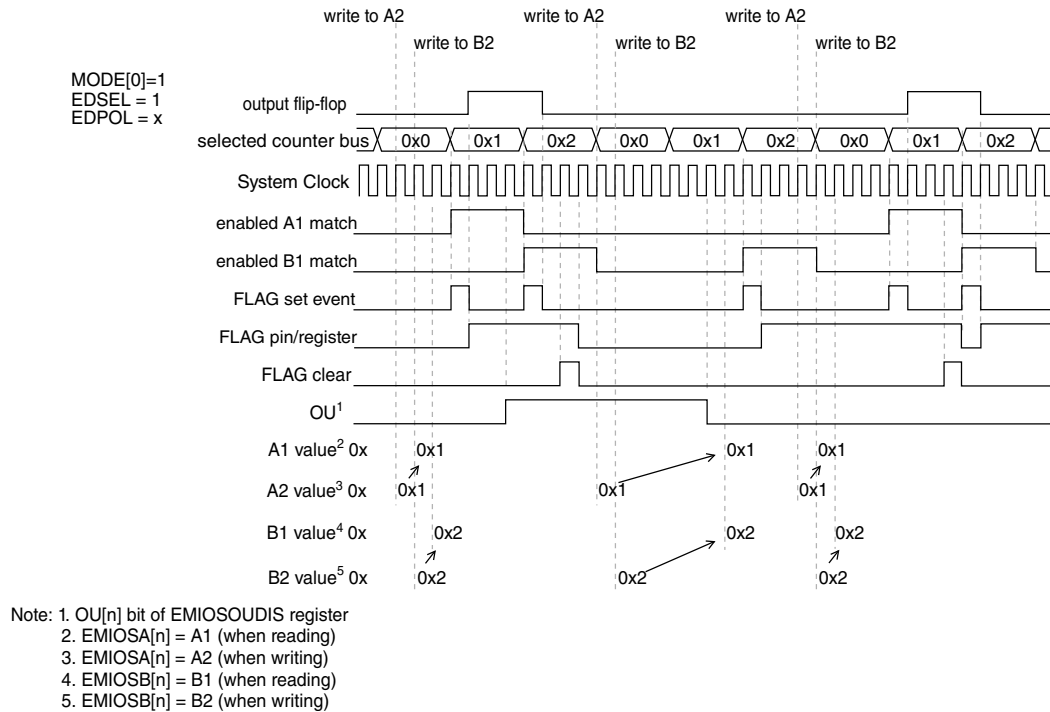


Figure 34-127. DAOC with transfer disabling example

### 34.6.1.1.7 Pulse/Edge Accumulation (PEA) Mode

The PEA mode returns the time taken to detect a desired number of input events. MODE[6] bit selects between continuous or single shot operation.

After writing to register A1, the internal counter is cleared on the first input event, ready to start counting input events and the selected timebase is latched into register B2. On the match between the internal counter and register A1, a counter bus capture is triggered to register A2 and B2. The data previously held in register B2 is transferred to register B1 and the FLAG bit is set to indicate that an event has occurred. The desired time interval can be determined by subtracting register B1 from A2. Registers EMIOSA[n] and EMIOSB[n] return the values in register A2 and B1, respectively.

As part of the coherency mechanism, reading EMIOSA[n] disables transfers from B2 to B1. These transfers are disabled until the next read of the EMIOS\_B\_n register. Reading the EMIOSB[n] register re-enables transfers from B2 to B1, to take effect at the next transfer event, as described above. <sup>1</sup>

In order to have coherent data in continuous mode of operation the following steps should be performed, assuming FLAG is initially cleared:

1. If B1 was not updated due to B2 to B1 transfer being disabled after reading register EMIOS\_A\_n, further EMIOS\_A\_n] and EMIOS\_B\_n reads will not return coherent data until a new bus capture is triggered to registers A2 and B2. This capture event is indicated by the channel FLAG being asserted. If enabled, the FLAG also generates an interrupt.

1. Wait for FLAG assertion;
2. Read EMIOSA[n] register;
3. Read EMIOSB[n] register;
4. Clear FLAG bit;
5. Return to step 1.

Accumulation cycles may be lost if the read is not performed in a timely manner. Whenever Overrun bit is asserted it means that one or more cycles have been lost.

Triggering of the counter clock (input event) is done by a rising or falling edge or both edges on the input pin. The polarity of the triggering edge is selected by the EDSEL and EDPOL bits in EMIOS\_C\_n register.

For continuous operation mode (MODE[6] cleared, MODE[0:6]=0001000), the counter is cleared on the next input event after a FLAG generation and continues to operate as described above.

For single shot operation (MODE[6] set, MODE[0:6]=0001001), the counter is not cleared or incremented after a FLAG generation, until a new writing operation to register A is performed.

Figure 34-128 and Figure 34-129 show how the Unified Channel can be used for continuous and single shot pulse/edge accumulation mode.

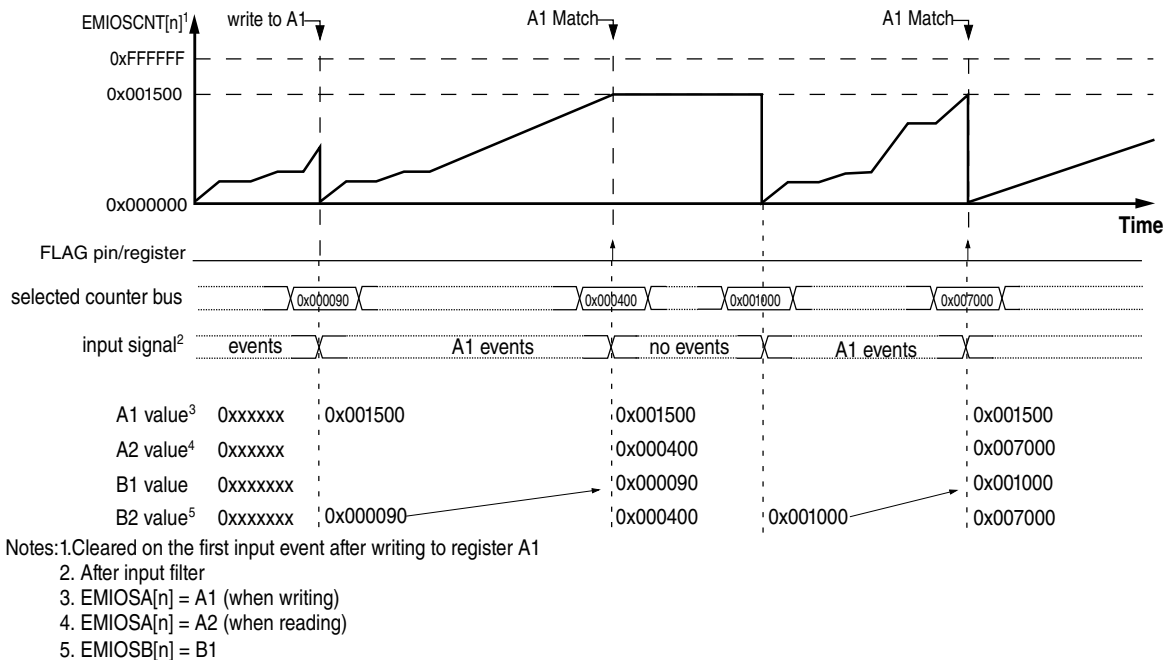


**Figure 34-128. Pulse/Edge Accumulation continuous mode example (MODE[6] bit is 0b)**

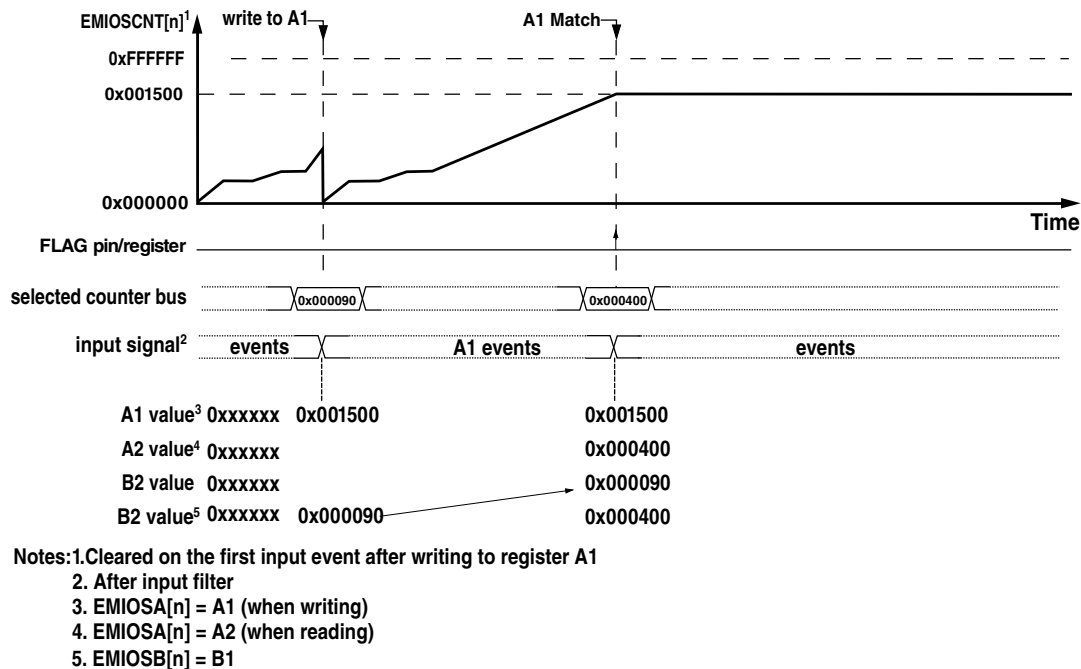


**Figure 34-129. Pulse/Edge Accumulation single-shot mode example (MODE[6] bit is 1b)**

### 34.6.1.1.8 Pulse/Edge Counting (PEC) Mode

The PEC mode returns the amount of pulses or edges detected on the input for a desired time window. MODE[6] bit selects between continuous or single shot operation.

Triggering of the internal counter is done by a rising or falling edge or both edges on the input signal. The polarity and the triggering edge is selected by EDSEL and EDPOL bits in EMIOSC[n] register.

Register A1 holds the start time and register B1 holds the stop time for the time window. After writing to register A1, when a match occur between comparator A and the selected timebase, the internal counter is cleared and it is ready to start counting input events. When the time base matches comparator B, the internal counter is disabled and its content is transferred to register A2. At the same time the FLAG bit is set. Reading registers EMIOSCNT[n] or A2 returns the amount of detected pulses.

For continuous operation (MODE[6] cleared, MODE[0:6]=0001010), the next match between comparator A and the selected time base clears the internal counter and counting is enabled again. In order to guarantee coherent measurements when reading EMIOSCNT[n] after the FLAG is set, the software must check if the time base value is out of the time interval defined by registers A1 and B1. Alternatively register A2 always holds the latest available measurement providing coherent data at any time after the first FLAG had occurred. This register is addressed by the alternate address EMIOSALTA[n].

For single shot operation (MODE[6] set, MODE[0:6]=0001011), the next match between comparator A and the selected time base has no effect, until a new write to register A is performed. The EMIOSCNT content is also transferred to register A2 when a match in the B comparator occurs.

Figure 34-130 and Figure 34-131 show how the Unified Channel can be used for continuous or single shot pulse/edge counting mode.

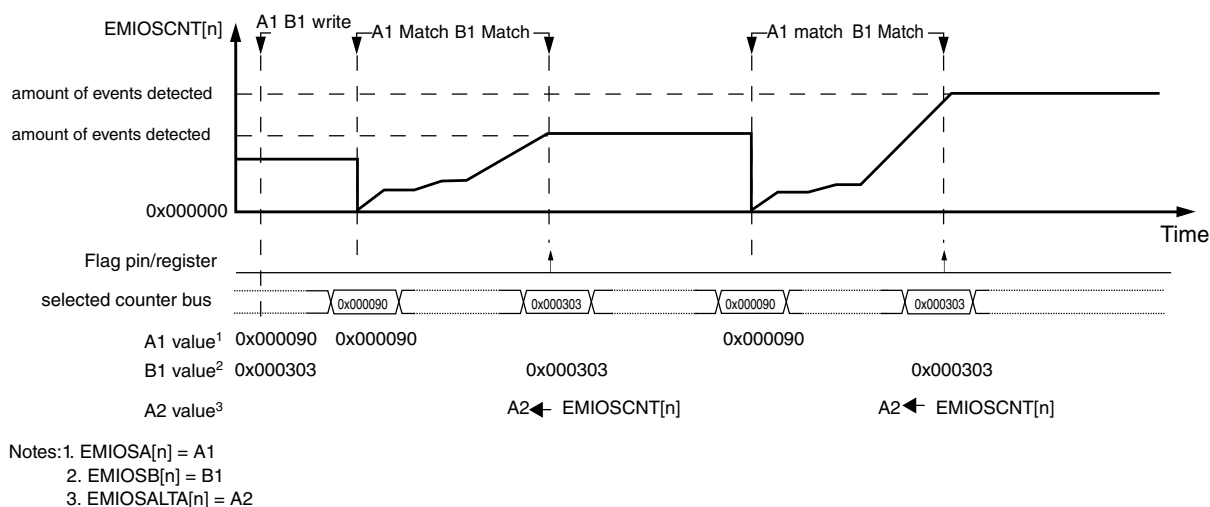


**Figure 34-130. Pulse/Edge Counting continuous mode example**

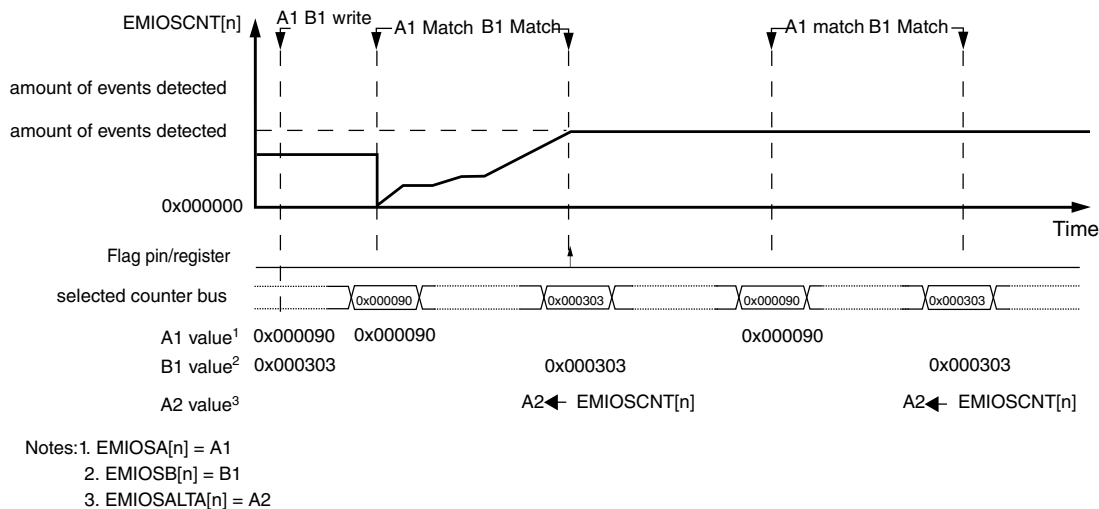


**Figure 34-131. Pulse/Edge Counting single-shot mode example**

### 34.6.1.1.9 Quadrature Decode (QDEC) Mode

Quadrature decode mode uses UC[n] operating in QDEC mode and the input programmable filter (IPF) from UC[n-1]. Note that UC[n-1] can be configured, at the same time, to an operation mode that does not use I/O pins, such as MC mode (modulus counter). The connection among the UCs is circular, i.e., when UC[0] is running in QDEC mode, the input programmable filter from UC[23] is being used, for the case of a 24-channel configuration.

This mode generates a FLAG every time the internal counter matches A1 register. The internal counter is automatically selected and is not cleared when entering this mode.

MODE[6] bit selects which type of encoder will be used: *count & direction* encoder or *phase\_A* and *phase\_B* encoders.

When operating with *count and direction* encoder (MODE[6] cleared), UC[n] input pin must be connected to the *direction* signal and UC[n-1] input pin must be connected to the *count* signal of the quadrature encoder. UC[n] EDPOL bit selects count direction according to *direction* signal and UC[n-1] EDPOL bit selects if the internal counter is clocked by the rising or falling edge of the *count* signal.

When operating with *phase\_A* and *phase\_B* encoder (MODE[6] set), UC[n] input pin must be connected to the *phase\_A* signal and UC[n-1] input pin must be connected to the *phase\_B* signal of the quadrature encoder. EDPOL bit selects the count direction according to the phase difference between *phase\_A* and *phase\_B* signals.

Figure 34-132 and Figure 34-133 show two Unified Channels configured to quadrature decode mode for *count & direction* encoder and *phase\_A* and *phase\_B* encoders, respectively.

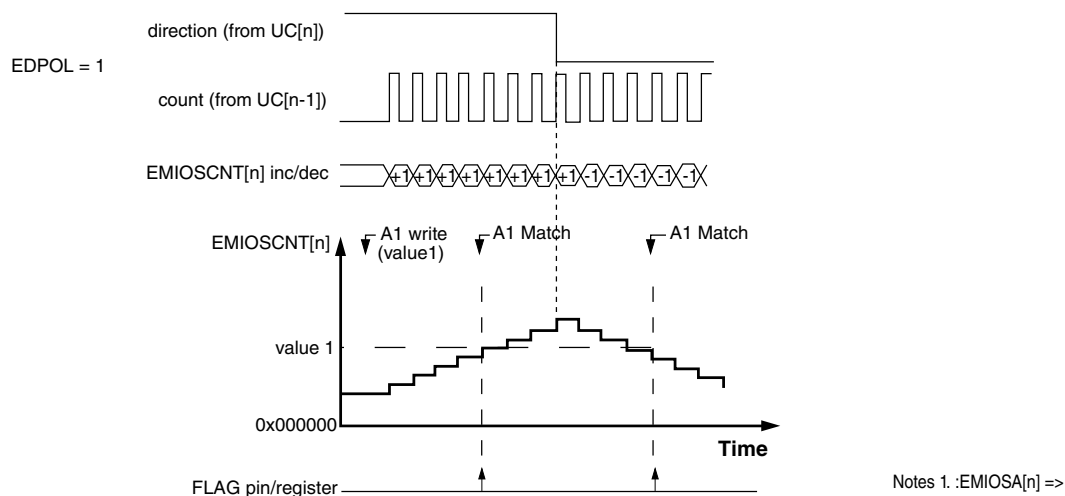


**Figure 34-132. Quadrature Decode mode example with *count and direction* encoder**



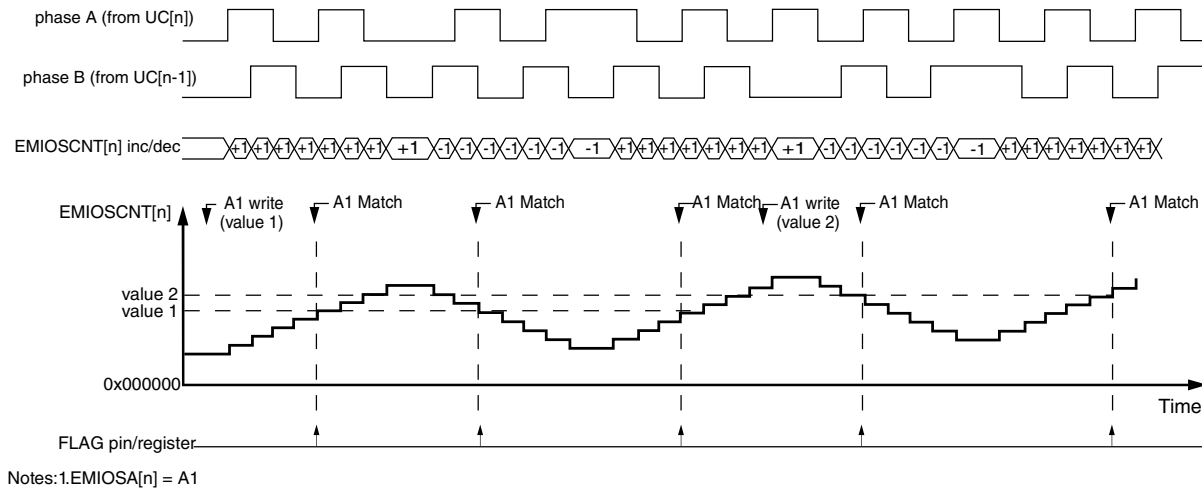


**Figure 34-133. Quadrature Decode mode example with *phase\_A* and *phase\_B* encoder**

#### 34.6.1.1.10 Windowed Programmable Time Accumulation (WPTA) Mode

The WPTA mode (MODE[0:6]=0001110) accumulates the sum of the total high time or low time of an input signal over a programmable interval (time window).

The prescaler bits UCPRE[0:1] in EMIOSC[n] register define the increment rate of the internal counter.

Register A1 holds the start time and register B1 holds the stop time of the programmable time interval. When a match occurs between register A and the selected timebase, the internal counter is cleared and it is ready to start counting. The internal counter is used as a time accumulator, i.e., it counts up when the input signal has the same polarity of EDPOL bit in EMIOSC[n] register and does not count otherwise. When a match occurs in comparator B, the internal counter is disabled regardless of the input signal polarity and the FLAG bit is set. At the same time the content of EMIOSCNT[n] is transferred to register A2. Reading registers EMIOSCNT[n] or A2 returns the high or low time of the input signal.

Note that EMIOSCNT[n] is stable only outside the time window defined from A1 to B1 matches, otherwise its content reflects a count in progress and not the final value. Alternatively to EMIOSCNT register A2 returns the latest available measurement. Since this register is updated only at comparator B matches it always contains stable and up-to-date data. In this mode this register is accessible through the alternate register address EMIOSALTA[n].

The following figure shows how the Unified Channel can be used to accumulate high time.

Timing diagram for the EMIOSCNT module. The diagram shows the relationship between the input signal, the time accumulator (EMIOSCNT), and the selected counter bus.

The input signal is shown as a square wave. The time accumulator (EMIOSCNT) is shown as a sawtooth waveform that increments during the A1 Match period and resets at the B1 Match.

The selected counter bus is shown as a series of values: 0x000100, 0x001500, 0x003000, and 0x004200.

The diagram also shows the A1 value, B1 value, and A2 value (A2 ← EMIOSCNT[n]) at various points in time.

Notes:

1. After the input filter
2. EMIOSA[n] = A1
3. EMIOSB[n] = B1
4. EMIOSALTA[n] = A2

- External clock is selected if MODE[0] is set. In this case the internal counter clears as soon as the match signal occurs. The channel FLAG is set at the same time the match occurs. Note that by having the internal counter cleared as soon as the match occurs and incremented at the next input event a shorter zero count is generated. See [Figure 34-160](#) and [Figure 34-161](#).
- Internal clock source is selected if MODE[0] is cleared. In this case the counter clears as soon as the match signal occurs. The channel FLAG is set at the same time the match occurs. At the next prescaler tick after the match the internal counter remains at zero and only resumes counting on the following tick. See [Figure 34-160](#) and [Figure 34-162](#).
- Internal counter clearing on match end (MODE[0:6]=001001b)
  - External clock is selected if MODE[0] is set. In this case the internal counter clears when the match signal is asserted and the input event occurs. The channel FLAG is set at the same time the counter is cleared. See [Figure 34-160](#) and [Figure 34-163](#).
  - Internal clock source is selected if MODE[0] is cleared. In this case the internal counter clears when the match signal is asserted and the prescaler tick occurs. The channel FLAG is set at the same time the counter is cleared. See [Figure 34-160](#) and [Figure 34-163](#).

### Note

If internal clock source is selected and the prescaler of the internal counter is set to 1 the MC mode behaves the same way even in Clear on Match Start or Clear on Match End sub-modes.

When in up/down count mode (MODE[0:6]=00101bb), a match between the internal counter and register A1 sets the FLAG and changes the counter direction from increment to decrement. A match between register B1 and the internal counter changes the counter direction from decrement to increment and sets the FLAG only if MODE[5] bit is set.

Only values different than 0x0 must be written at A register. Loading 0x0 leads to unpredictable results.

Updates on register A or counter in MC mode may cause loss of match in the current cycle if the transfer occurs near the match. In this case, the counter may rollover and resume operation in the next cycle.

Register B2 has no effect in MC mode. Nevertheless, register B2 can be accessed for reads and writes by addressing EMIOSB.

The following two figures show how the Unified Channel can be used as modulus counter in up mode and up/down mode, respectively.

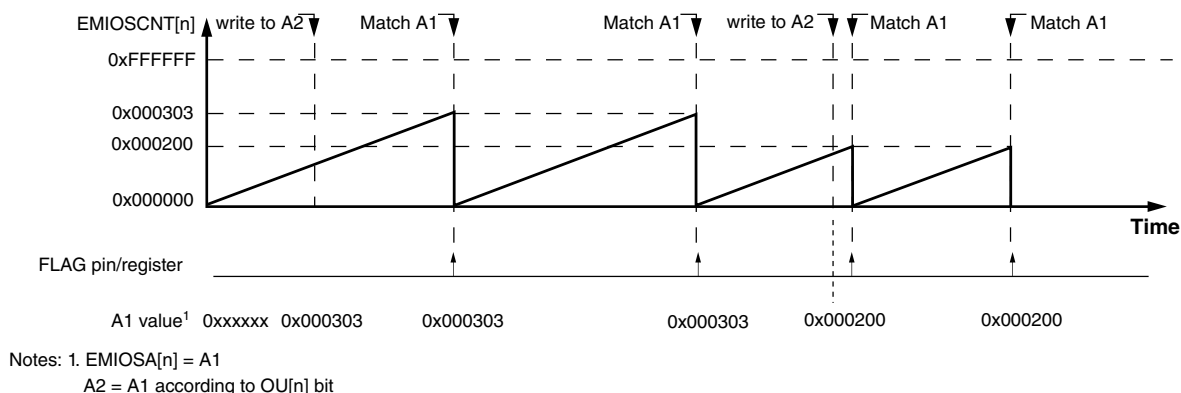


**Figure 34-135. Modulus Counter up mode example (MODE[4] bit is 0)**

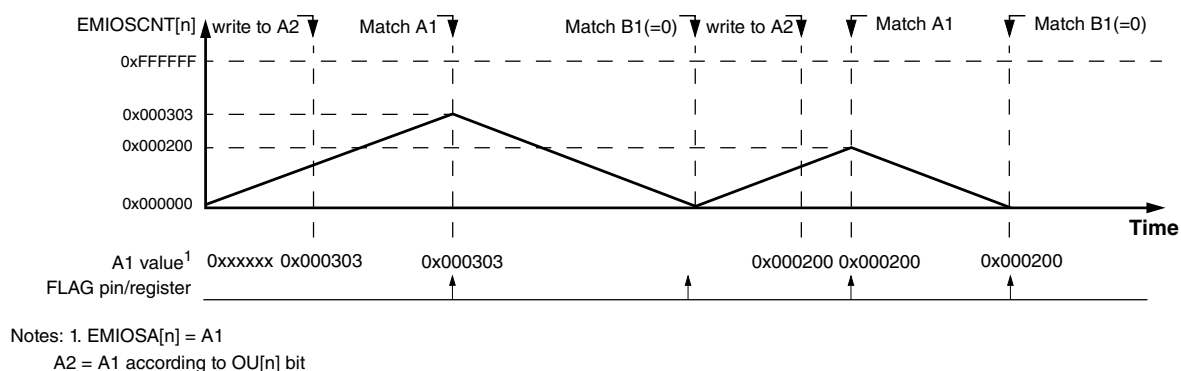


**Figure 34-136. Modulus Counter up/down mode example (MODE[4] bit is 1)**

## 34.6.1.1.12 Modulus Counter Buffered (MCB) Mode

The MCB mode provides a time base which can be shared with other channels through the internal counter buses. Register A1 is double buffered thus allowing smooth transitions between cycles when changing A2 register value on the fly. Register A1 is updated at the cycle boundary, which is defined as when the internal counter transitions to 0x1.

The internal counter values operates within a range from 0x1 up to register A1 value. If when entering MCB mode coming out from GPIO mode the internal counter value is not within that range then the A match will not occur causing the channel internal counter to wrap at the maximum counter value which is 0xff\_ffff for a 24-bit counter. After the counter wrap occurs it returns to 0x1 and resume normal MCB mode operation. Thus in order to avoid the counter wrap condition make sure its value is within the 0x1 to register A1 value range when the MCB mode is entered.

MODE[6] bit selects internal clock source if cleared or external if set. When external clock is selected the input channel pin is used as the channel clock source. The active edge of this clock is defined by EDPOL and EDSEL bits in the EMIOSC[n] channel register.

When entering in MCB mode, if up counter is selected by MODE[4]=0 (MODE[0:6]=101000b), the internal counter starts counting from its current value to up direction until A1 match occurs. The internal counter is set to 0x1 when its value matches A1 value and a clock tick occurs (either prescaled clock or input pin event).

If up/down counter is selected by setting MODE[4]=1, the counter changes direction at A1 match and counts down until it reaches the value 0x1. After it has reached 0x1 it is set to count in up direction again. Register B1 is used to generate a match in order to set the internal counter in up-count direction if up/down mode is selected. Register B1 cannot be changed while this mode is selected.

Note that differently from the MC mode, the MCB mode counts between 0x1 and A1 register value. Only values greater than 0x1 must be written at register A1. Loading values other than those leads to unpredictable results. The counter cycle period is equal to A1 value in up counter mode. If in up/down counter mode the period is defined by the expression:  $(2 * A1) - 2$ .

Figure 34-137 describes the counter cycle for several A1 values. Register A1 is loaded with A2 register value at the cycle boundary. Thus any value written to A2 register within cycle **n** will be updated to A1 at the next cycle boundary and therefore will be used on cycle **n+1**. The cycle boundary between cycle **n** and cycle **n+1** is defined as when the internal counter transitions from A1 value in cycle **n** to 0x1 in cycle **n+1**. Note that the FLAG is generated at the cycle boundary and has a synchronous operation, meaning that it is asserted one system clock cycle after the FLAG set event.

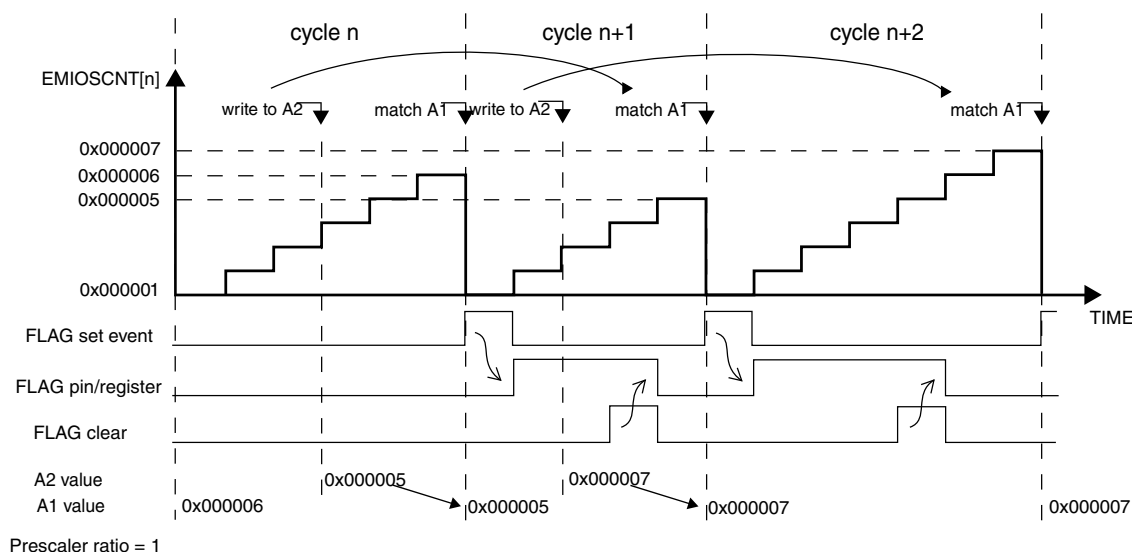


**Figure 34-137. Modulus Counter Buffered (MCB) Up Count mode**

Figure 34-138 describes the MCB in up/down counter mode (MODE[0:6]=10101bb). Register A1 is updated at the cycle boundary. If A2 is written in cycle *n*, this new value will be used in cycle *n+1* for A1 match. Flags are generated only at A1 match start if MODE[5] is 0. If MODE[5] is set to 1 flags are also generated at the cycle boundary.

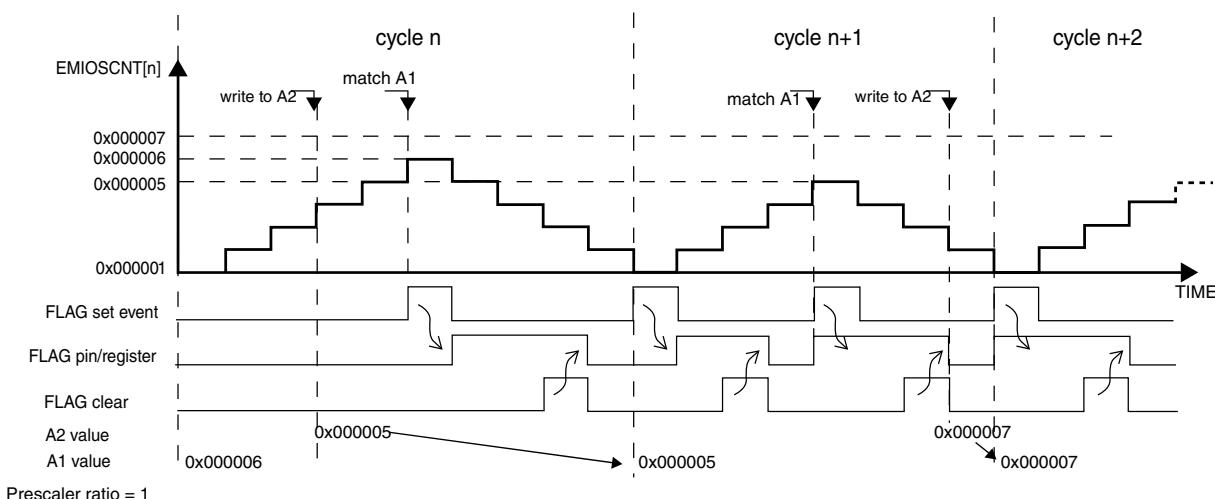


**Figure 34-138. Modulus Counter Buffered (MCB) Up/Down Mode**

The figure below describes in more detail, the register A1 update process in up counter mode. The A1 load signal is generated at the last system clock period of a counter cycle. Thus, A1 is updated with A2 value at the same time that the counter (EMIOSCNT[n]) is loaded with 0x1. The load signal pulse has the duration of one system clock period. If A2 is written within cycle *n* its value is available at A1 at the first clock of cycle *n+1* and the new value is used for match at cycle *n+1*. The update disable bits OU[n] of EMIOSOUDIS register can be used to control the update of this register, thus allowing to delay the A1 register update for synchronization purposes.

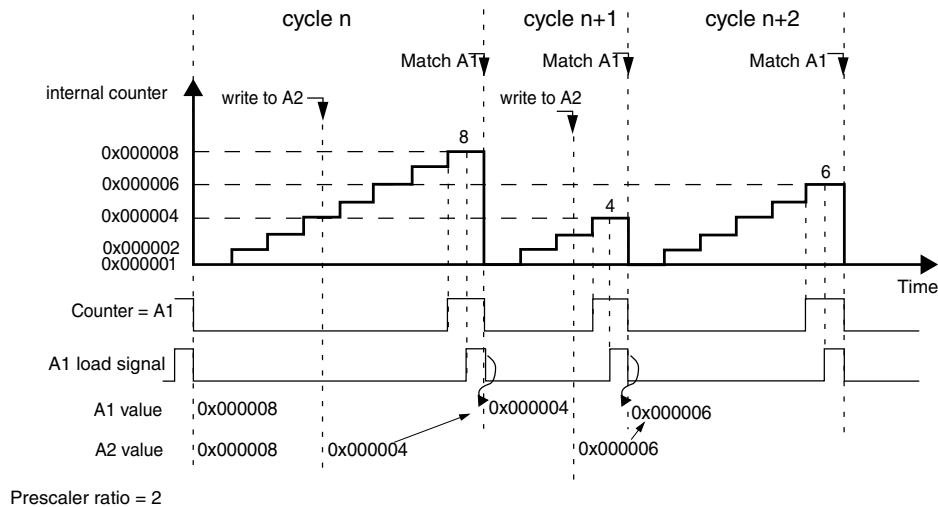


**Figure 34-139. MCB Mode Register A1 Update in Up Counter Mode**

The figure below describes the register A1 update in up/down counter mode. Note that A2 can be written at any time within cycle **n** in order to be used in cycle **n+1**. Thus A1 receives this new value at the next cycle boundary. Note that the update disable bits **OU[n]** of **EMIOSOUDIS** register can be used to disable the update of register A1.

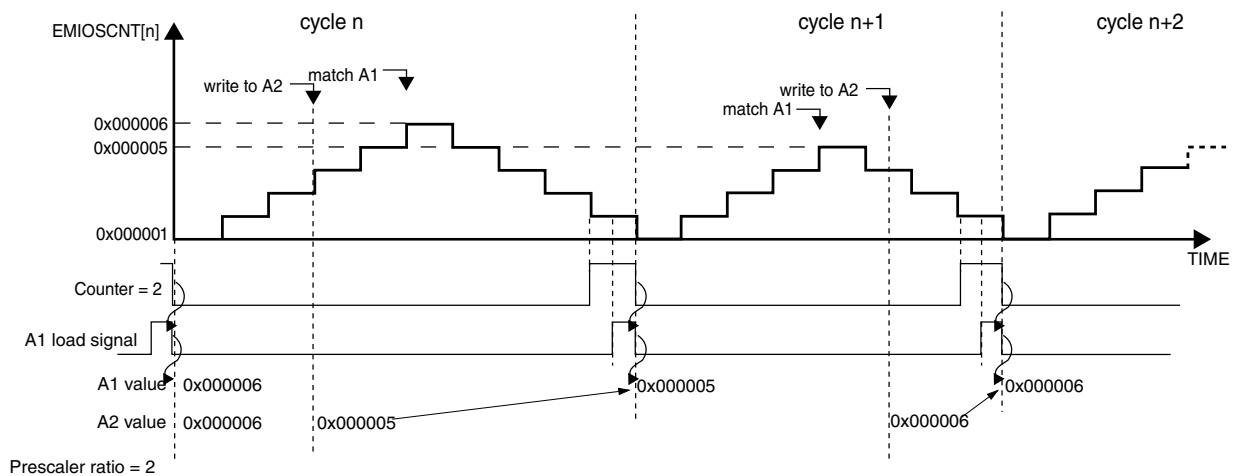


**Figure 34-140. MCB Mode Register A1 Update in Up/Down Counter Mode**

### 34.6.1.1.13 Output Pulse Width and Frequency Modulation (OPWFM) Mode

In this mode, duty cycle of output signal is the value defined in register A1 plus one and the period is the value defined in register B1 plus one. **MODE[6]** bit controls the transfer from register B2 to B1, which can be done either immediately (**MODE[6]** cleared, **MODE[0:6]=00110b0**), providing the fastest change in the duty cycle, or at every match of register A1 (**MODE[6]** set, **MODE[0:6]=00110b1**).

When OPWFM mode is entered, coming out from GPIO mode, the output flip-flop is set to the complement of the **EDPOL** bit in the **EMIOSC[n]** register.

The internal counter is automatically selected as a time base, therefore the BSL[0:1] bits in register EMIOSC[n] has no meaning. When a match on comparator A occurs, the output flip-flop is set to the value of the EDPOL bit. When a match occurs on comparator B, the output flip-flop is set to the complement of the EDPOL bit and the internal counter is cleared.

A FLAG can be generated at match B, when MODE[5] is cleared, or in both matches, when MODE[5] is set.

At any time, the FORCMA and FORCMB bits allow the software to force the output flip-flop to the level corresponding to a match on A or B respectively. Also, FORCMB clears the internal counter. Note that the FLAG bit is not set by the FORCMA or FORCMB operations.

If subsequent comparisons occur on comparators A and B, the PWFM pulses continue to be output, regardless of the state of the FLAG bit.

In order to achieve 100% duty cycle, both registers A1 and B1 must be set to the same value. When a simultaneous match occurs on comparators A and B, the output flip-flop is set to the value of EDPOL bit. 0% duty cycle is possible by writing 0x0 to register A. When a match occurs, the output flip-flop is set at every period to the complement of EDPOL bit. The transfer from register B2 to B1 is still controlled by MODE[6] bit.

### Note

Writing 0x0 to A1 and B1 produces a duty cycle of 0%.

Figure 34-141 shows the Unified Channel running in OPWFM mode with immediate register update and Figure 34-142 shows the Unified Channel running in OPWFM mode with next period update.

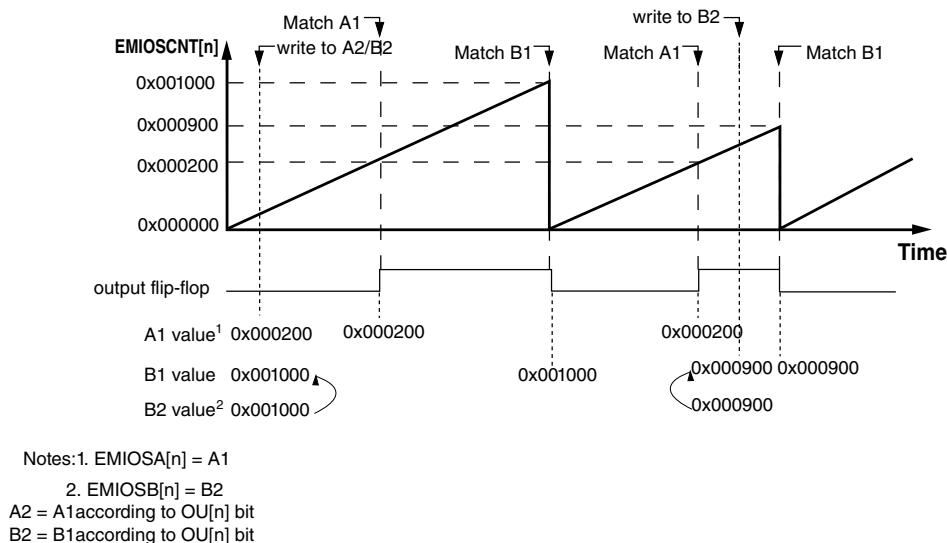


**Figure 34-141. OPWFM with immediate update (MODE[6] bit is 0b)**



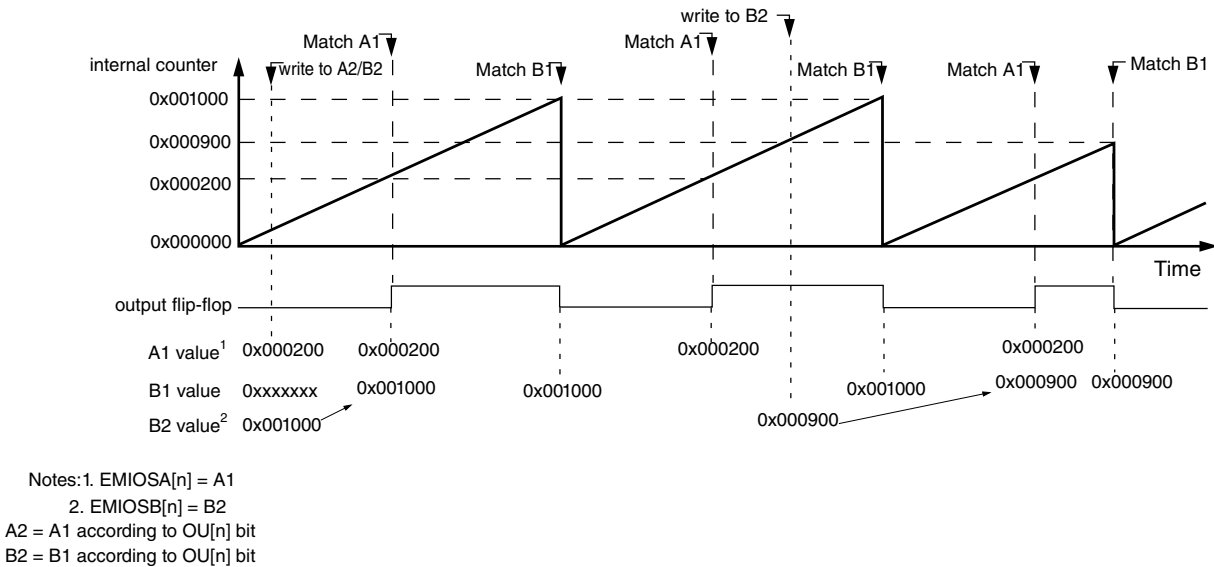


**Figure 34-142. OPWFM with next period update (MODE[6] bit is 1b)**

#### 34.6.1.1.14 Output Pulse Width and Frequency Modulation Buffered (OPWFMB) Mode

This mode (MODE[0:6]=10110b0) provides waveforms with variable duty cycle and frequency. The internal channel counter is automatically selected as the time base when this mode is selected. Register A1 indicates the duty cycle and B1 register the frequency. Both registers A1 and B1 are double buffered to allow smooth signal generation when changing the registers values.. 0% and 100% duty cycles are supported.

At OPWFMB mode entry, the output flip-flop is set to the value of the EDPOL bit in the EMIOSC[n] register.

In order to provide smooth and consistent channel operation this mode differs substantially from the OPWFM mode. The main differences reside in the A1 and B1 registers update, on the delay from the A1 match to the output pin transition and on the range of the internal counter values which starts from 0x1 up to register B1 value.

When entering OPWFMB mode coming out of GPIO mode, the internal counter value is not within that range then the B match will not occur causing the channel internal counter to wrap at the maximum counter value which is 0xff\_fff for a 24-bit counter. After the counter wrap occurs it returns to 0x1 and resume normal OPWFMB mode operation. Thus in order to avoid the counter wrap condition make sure its value is within the 0x1 to B1 register value range when the OPWFMB mode is entered.

When a match on comparator A occurs the output register is set to the value of EDPOL. When a match on comparator B occurs the output register is set to the complement of EDPOL. B1 match also causes the internal counter to transition to 0x1, thus restarting the counter cycle.

Only values greater than 0x1 are allowed to be written to register B1. Loading values other than those leads to unpredictable results.

Figure 34-143 describes the operation of the OPWFMB mode regarding output pin transitions and A1/B1 registers match events. Note that the output pin transition occurs when the A1 or B1 match signal is deasserted which is indicated by the A1 match negedge detection signal. If register A1 is set to 0x4 the output pin transitions 4 counter periods after the cycle had started, plus one system clock cycle. Note that in the example shown in Figure 34-143 the internal counter prescaler has a ratio of two.

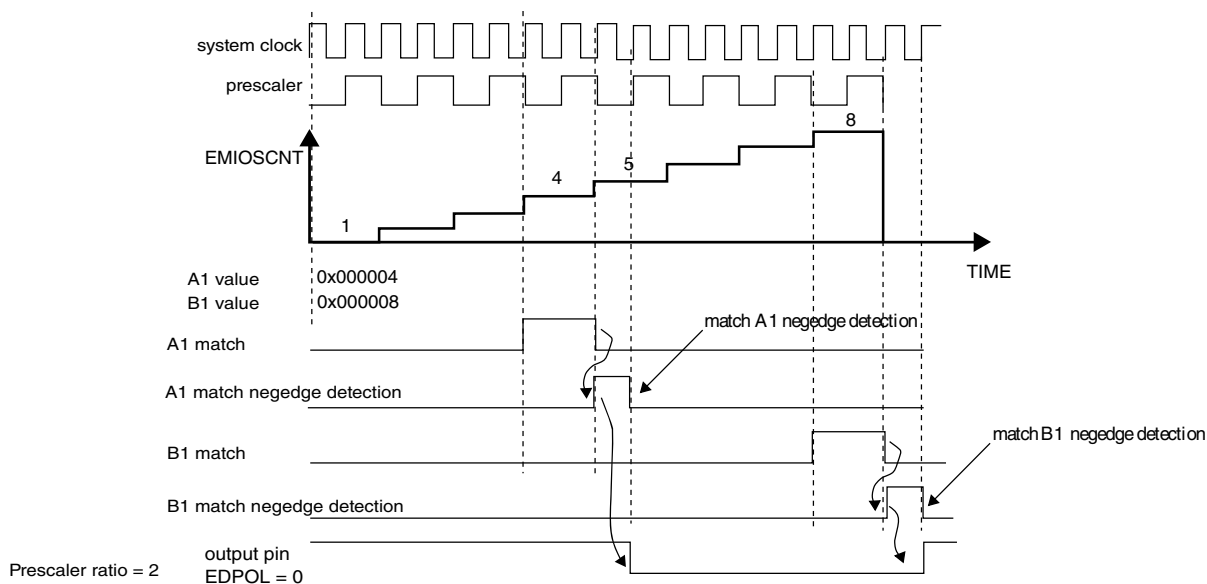


**Figure 34-143. OPWFMB A1 and B1 match to Output Register Delay**

Figure 34-144 describes the generated output signal if A1 is set to 0x0. Since the counter does not reach zero in this mode, the channel internal logic infers a match as if A1=0x1 with the difference that in this case, the posedge of the match signal is used to trigger the output pin transition instead of the negedge used when A1=0x1. Note that A1 posedge match signal from cycle  $n+1$  occurs at the same time as B1 negedge match signal from cycle  $n$ . This allows to use the A1 posedge match to mask the B1 negedge match when they occur at the same time. The result is that no transition occurs on the output flip-flop and a 0% duty cycle is generated.

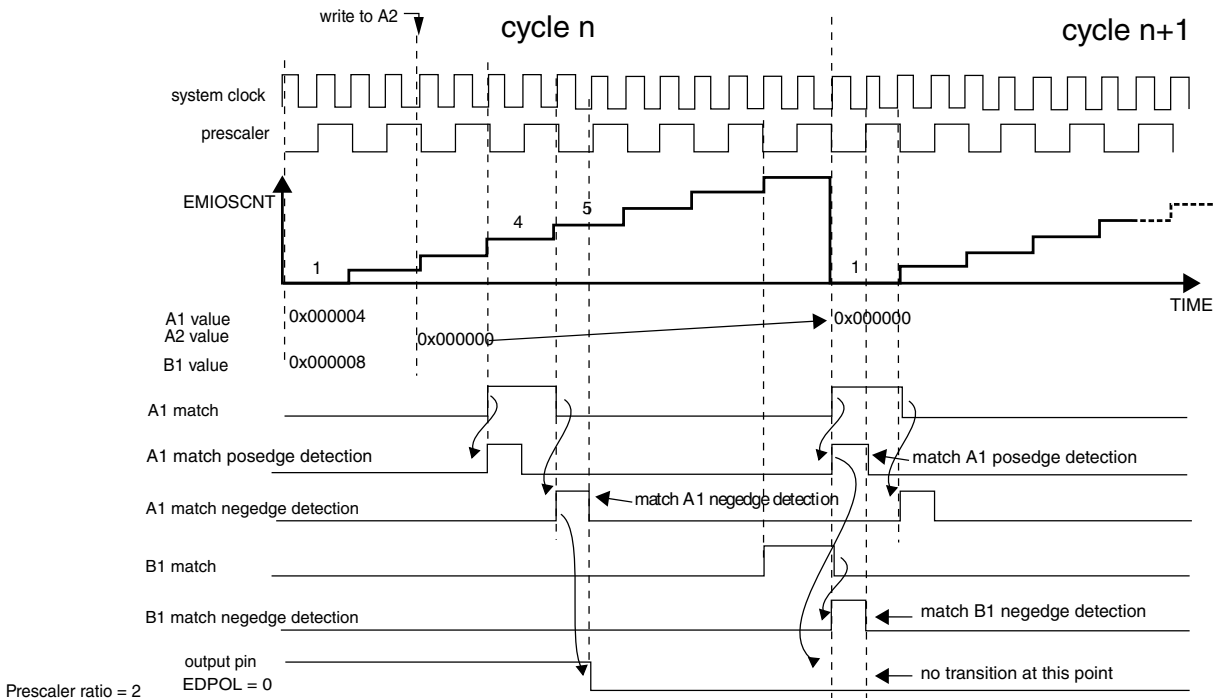


**Figure 34-144. OPWFMB Mode with A1 = 0 (0% duty cycle)**

Figure 34-145 describes the timing for the A1 and B1 registers load. The A1 and B1 load use the same signal which is generated at the last system clock period of a counter cycle. Thus, A1 and B1 are updated respectively with A2 and B2 values at the same time that the counter (EMIOSCNT[n]) is loaded with 0x1. This event is defined as the cycle boundary. The load signal pulse has the duration of one system clock period. If A2 and B2 are written within cycle **n** their values are available at A1 and B1, respectively, at the first clock of cycle **n+1** and the new values are used for matches at cycle **n+1**. The update disable bits OU[n] of EMIOSOUDIS register can be used to control the update of these registers, thus allowing to delay the A1 and B1 registers update for synchronization purposes.

In Figure 34-145, it is assumed that both the channel and global prescalers are set to 0x1 (each divide ratio is two), meaning that the channel internal counter transitions at every four system clock cycles. FLAGS can be generated only on B1 matches when MODE[5] is cleared, or on both A1 and B1 matches when MODE[5] is set. Since B1 flag occurs at the cycle boundary, this flag can be used to indicate that A2 or B2 data written on cycle **n** were loaded to A1 or B1, respectively, thus generating matches in cycle **n+1**. Note that the FLAG has a synchronous operation, meaning that it is asserted one system clock cycle after the FLAG set event.

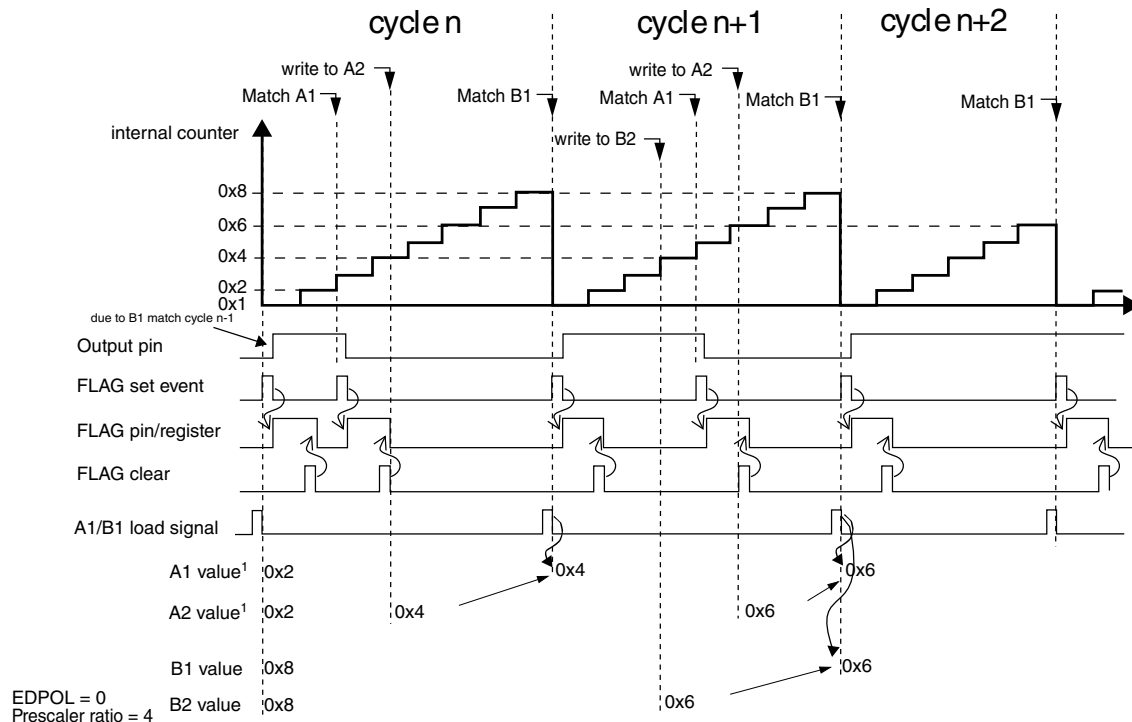


**Figure 34-145. OPWFMB A1 and B1 Registers Update and Flags (MODE6] bit is 0b)**

Figure 34-146 describes the operation of the Output Disable feature in OPWFMB mode. Differently from the OPWFM mode, the output disable forces the channel output flip-flop to EDPOL bit value. This functionality targets applications that use active high signals and a high to low transition at A1 match. In this case EDPOL should be set to 0. Note that both the channel and global prescalers are set to 0x0 (each divide ratio is one), meaning that the channel internal counter transitions at every system clock cycle.

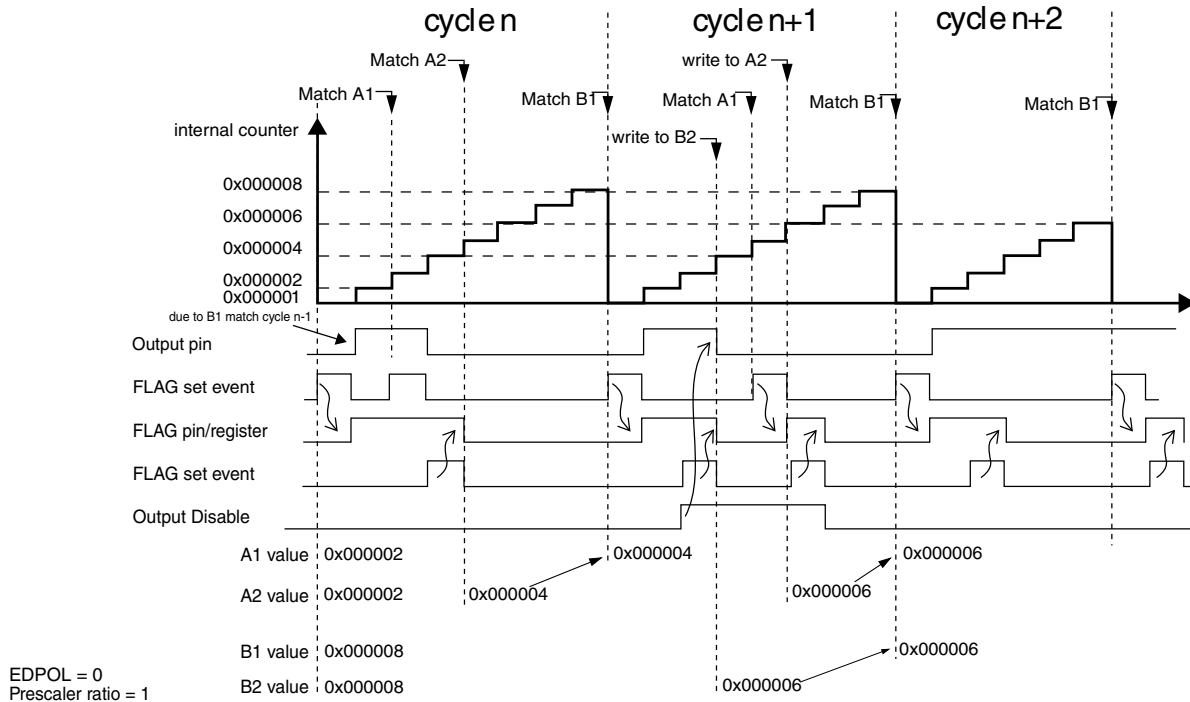


**Figure 34-146. OPWFMB Mode with Active Output Disable (MODE[6] bit is 1b)**

Note that the output disable has a synchronous operation, meaning that the assertion of the Output Disable input pin causes the channel output flip-flop to transition to EDPOL at the next system clock cycle. If the Output Disable input is deasserted the output pin transition at the following A1 or B1 match.

In [Figure 34-146](#), it is assumed that the Output Disable input is enabled and selected for the Channel. Please, refer to UC Control Register (EMIOSCn) for a detailed description about the ODIS and ODISSL bits, respectively enable and selection of the Output Disable inputs.

The FORCMA and FORCMB bits allow the software to force the output flip-flop to the level corresponding to a match on comparators A or B respectively. Similarly to a B1 match FORCMB sets the internal counter to 0x1. The FLAG bit is not set by the FORCMA or FORCMB bits being asserted.

[Figure 34-147](#) describes the generation of 100% and 0% duty cycle signals. It is assumed EDPOL = 0 and the resultant prescaler value is 1. Initially, A1=0x8 and B1=0x8. In this case, B1 match has precedence over A1 match, thus the output flip-flop is set to the complement of EDPOL bit. This cycle corresponds to a 100% duty cycle signal. The same output signal can be generated for any A1 value greater or equal to B1.

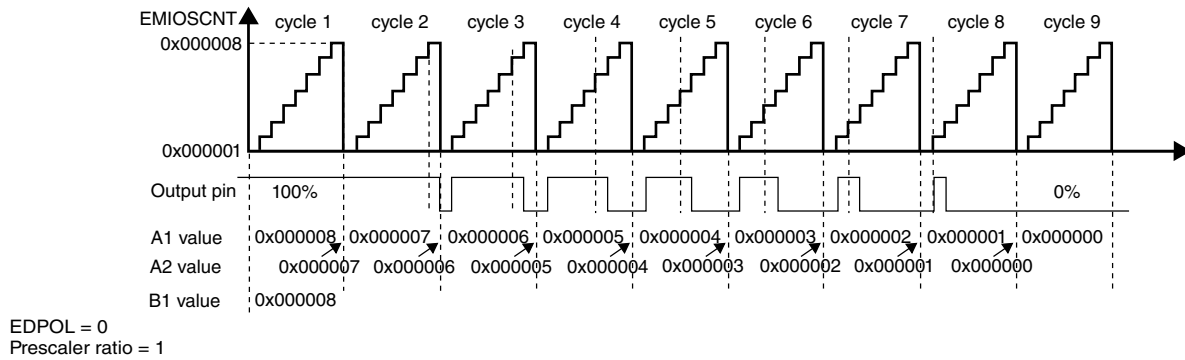


**Figure 34-147. OPWFMB Mode from 100% to 0% Duty Cycle**

A 0% duty cycle signal is generated if A1=0x0 as shown in [Figure 34-147](#) cycle 9. In this case B1=0x8 match from cycle 8 occurs at the same time as the A1=0x0 match from cycle 9. Please, refer to [Figure 34-144](#) for a description of the A1 and B1 match generation. In this case A1 match has precedence over B1 match and the output signal transitions to EDPOL.

#### 34.6.1.1.15 Center Aligned Output Pulse Width Modulation with Dead Time Insertion (OPWMC) Mode

This operation mode generates a center aligned PWM with dead time insertion in the leading (MODE[0:6]=00111b1) or trailing edge (MODE[0:6]=00111b0).

The selected counter bus must be running an up/down time base, as shown in the 'Modulus Counter up/down mode example' figure. BSL[0:1] bits select the time base. Register A1 contains the ideal duty cycle for the PWM signal and is compared with the selected time base. Register B1 contains the dead time value and is compared with the internal counter. For a leading edge dead time insertion, the output PWM duty cycle is equal to the difference between register A1 and register B1, and for a trailing edge dead time insertion, the output PWM duty cycle is equal to the sum of register A1 and register B1. Mode[6] bit selects between trailing and leading dead time insertion, respectively.

#### Note

The internal counter may be running in the internal prescaler ratio, while the selected time base may be running in a different prescaler ratio. The output signal may produce an unexpected output if the dead time interval is greater than the duty cycle of the PWM signal.

When OPWMC mode is entered, coming out from GPIO mode, the output flip-flop is set to the complement of the EDPOL bit in the EMIOSC[n] register.

When operating with leading edge dead time insertion, the first match between A1 and the selected time base clears the internal counter and switches the selected time base to the internal counter. When a match occurs between register B1 and the selected time base, the output flip-flop is set to the value of the EDPOL bit and the time base is switched to the selected counter bus. In the next match between register A1 and the selected time base, the output flip-flop is set to the complement of the EDPOL bit. This sequence repeats continuously.

When operating with trailing edge dead time insertion, the first match between A1 and the selected time base sets the output flip-flop to the value of the EDPOL bit. In the next match between register A1 and the selected time base, the internal counter is cleared and the selected time base is switched to the internal counter. When a match occurs between register B1 and the selected time base, the output flip-flop is set to the complement of the EDPOL bit and the time base is switched to the selected counter bus. This sequence repeats continuously.

FLAG can be generated in the trailing edge of the output PWM signal when MODE[5] is cleared, or in both edges, when MODE[5] is set.

At any time, the FORCMA or FORCMB bits are equivalent to a successful comparison on comparator A or B with the exception that the FLAG bit is not set.

### Note

When in freeze state, the FORCMA or FORCMB bits only allow the software to force the output flip-flop to the level corresponding of a match on A or B respectively.

If subsequent matches occur on comparators A and B, the PWM pulses continue to be generated, regardless of the state of the FLAG bit.

In order to achieve a duty cycle of 100%, both registers A1 and B1 must be set to the same value. When a simultaneous match occurs between the selected time base and registers A1 and B1, the output flip-flop is set to the value of EDPOL bit and the selected time base switches to the selected counter bus, allowing a new cycle to begin at any time, as previously described. 0% duty cycle is possible by writing 0x0 to register A. When a match occurs, the output flip-flop is set to the complement of EDPOL bit and the selected time base switches to the selected counter bus, allowing a new cycle to begin at any time, as previously described. In both cases, FLAG is generated regardless of MODE[6] bit.

### Note

If A1 and B1 are set to the 0x0, a 0% duty cycle waveform is produced.

Figure 34-148 and Figure 34-149 show the Unified Channel running in OPWMC with leading and trailing dead time, respectively.

## Functional Description

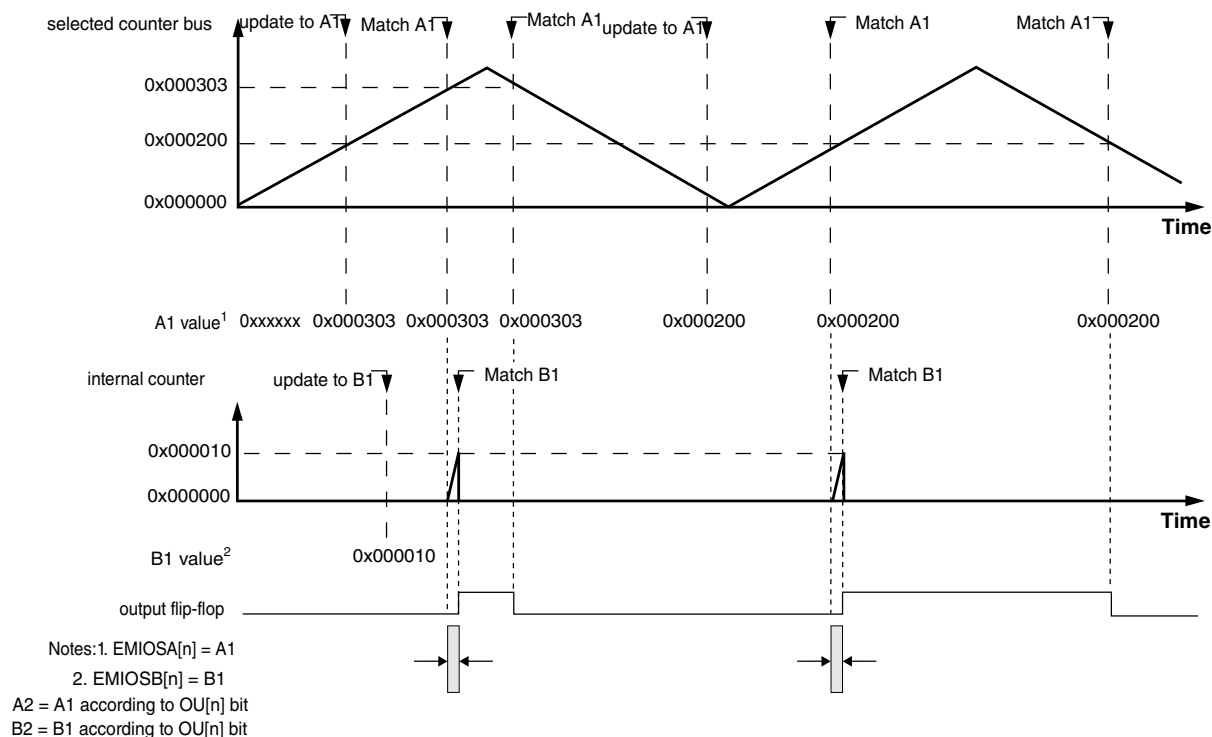


**Figure 34-148. OPWMC with leading dead time insertion (MODE [6] bit is 1b)**

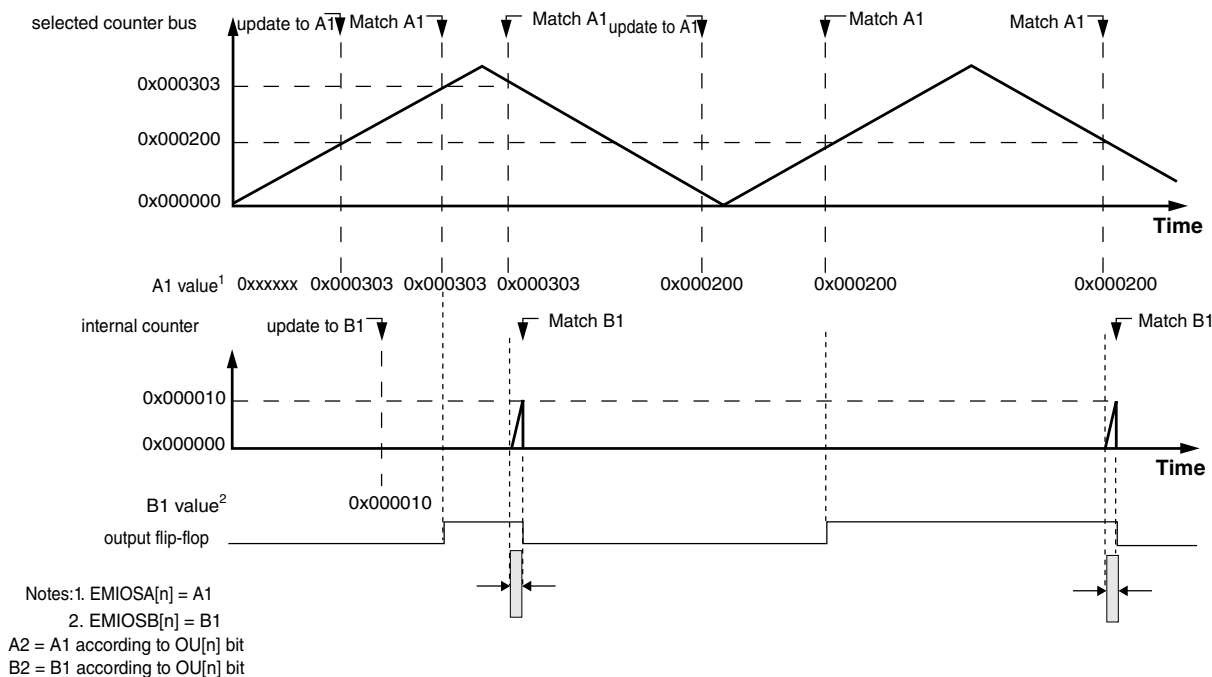


**Figure 34-149. OPWMC with trailing dead time insertion (MODE [6] bit is 0b)**



### 34.6.1.1.16 Output Pulse Width Modulation (OPWM) Mode

Registers A1 and B1 define the leading and trailing edges of the PWM output pulse, respectively. MODE[6] bit controls the transfer from register B2 to B1, which can be done either immediately (MODE[6] cleared, MODE[0:6]=01000b0), providing the fastest change in the duty cycle, or at every match of register A1 (MODE[6] set, MODE[0:6]=01000b1).

The value loaded in register A1 is compared with the value on the selected time base. When a match on comparator A occurs, the output flip-flop is set to the value of the EDPOL bit. When a match occurs on comparator B, the output flip-flop is set to the complement of the EDPOL bit.

A FLAG can be generated at match B, when MODE[5] is cleared, or in both matches, when MODE[5] is set.

At any time, the FORCMA and FORCMB bits allow the software to force the output flip-flop to the level corresponding to a match on A or B respectively. Note that FLAG bit is not set by the FORCMA and FORCMB operations.

If subsequent matches occur on comparators A and B, the PWM pulses continue to be generated, regardless of the state of the FLAG bit.

At OPWM mode entry the output flip-flop is set to the complement of the EDPOL bit in the EMIOSC[n] register.

In order to achieve 100% duty cycle, both registers A1 and B1 must be set to the same value. When a simultaneous match on comparators A and B occur, the output flip-flop is set at every period to the value of EDPOL bit. 0% duty cycle is possible by writing 0x0 to register A. When a match occurs, the output flip-flop is set at every period to the complement of EDPOL bit. The transfer from register B2 to B1 is still controlled by the MODE[6] bit.

#### Note

If A1 and B1 are set to the 0x0, a 0% duty cycle waveform is produced.

[Figure 34-150](#) and [Figure 34-151](#) show the Unified Channel running in OPWM with immediate update and next period update, respectively.

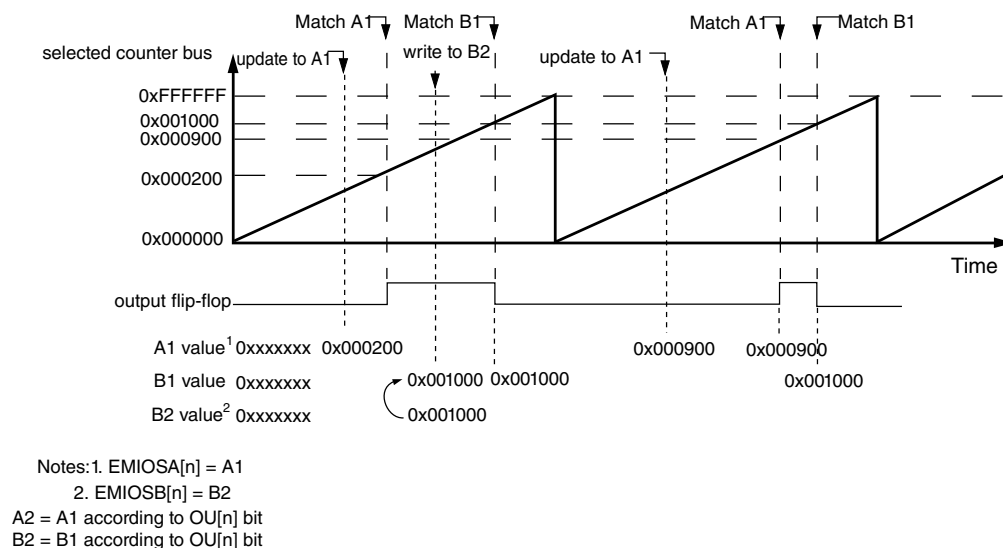


**Figure 34-150. PWM with immediate update (MODE[6] is 0b)**

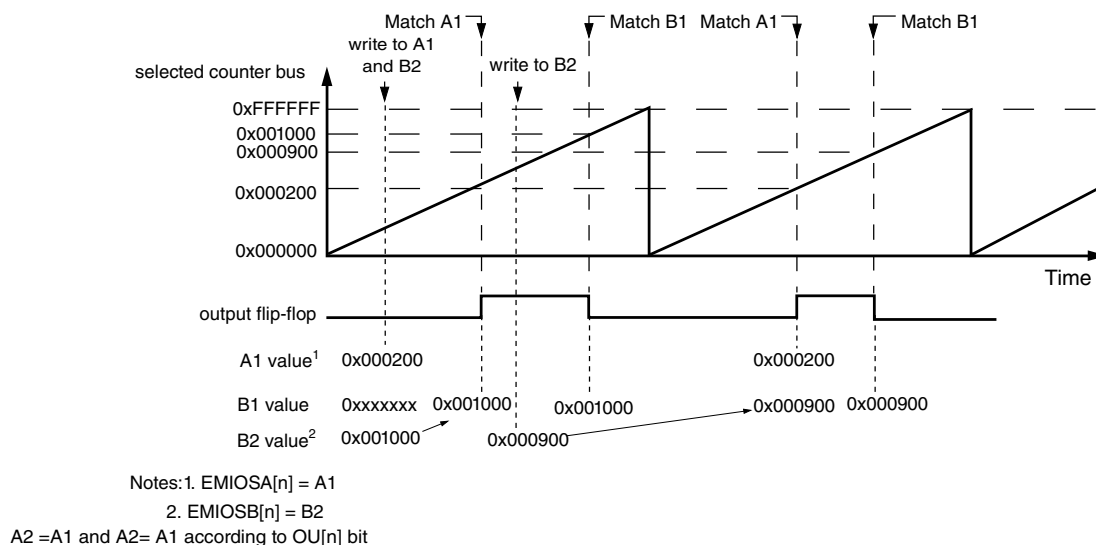


**Figure 34-151. PWM with next period update (MODE[6] is 1b)**

### 34.6.1.1.17 Output Pulse Width Modulation Buffered (OPWMB) Mode

OPWMB mode (MODE[0:6]=11000b0) is used to generate pulses with programmable leading and trailing edge placement. An external counter driven in MCB Up mode must be selected from one of the counter buses. A1 register value defines the first edge and B1 the second edge. The output signal polarity is defined by the EDPOL bit. If EDPOL is zero, a negative edge occurs when A1 matches the selected counter bus and a positive edge occurs when B1 matches the selected counter bus.

The A1 and B1 registers are double buffered and updated from A2 and B2, respectively, at the cycle boundary. The load operation is similar to the OPWFMB mode. Please refer to [Figure 34-145](#) for more information about A1 and B1 registers update.

FLAG can be generated at B1 matches, when MODE[5] is cleared, or in both A1 and B1 matches, when MODE[5] is set. If subsequent matches occur on comparators A and B, the PWM pulses continue to be generated, regardless of the state of the FLAG bit.

FORCMA and FORCMB bits allow the software to force the output flip-flop to the level corresponding to a match on A1 or B1 respectively. The FLAG bit is not set by the FORCMA and FORCMB operations.

At OPWMB mode entry the output flip-flop is set to the value of the EDPOL bit in the EMIOSC[n] register.

Some rules applicable to the OPWMB mode are:

- B1 matches have precedence over A1 matches if they occur at the same time within the same counter cycle
- A1=0 match from cycle **n** has precedence over B1 match from cycle **n-1**
- A1 matches are masked out if they occur after B1 match within the same cycle
- Any value written to A2 or B2 on cycle **n** is loaded to A1 and B1 registers at the following cycle boundary (assuming OU[n] bit of EMIOSOUDIS register is not asserted). Thus the new values will be used for A1 and B1 matches in cycle **n+1**

The following figure describes the operation of the OPWMB mode regarding A1 and B1 matches and the transition of the channel output pin. In this example EDPOL is set to zero.

## Functional Description

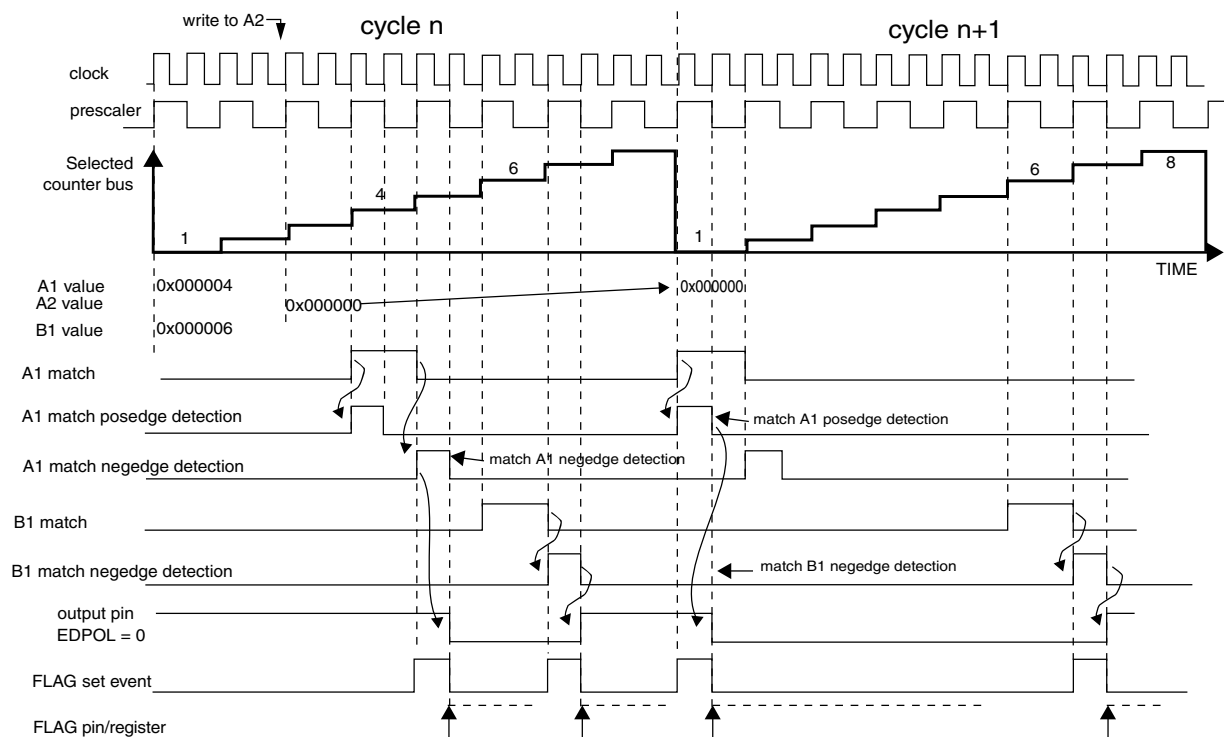


**Figure 34-152. OPWMB Mode Matches and Flags**

Note that the output pin transitions are based on the negedges of the A1 and B1 match signals. The figure above shows in cycle *n+1* the value of A1 register being set to zero. In this case the match posedge is used instead of the negedge to transition the output flip-flop.

The figure below describes the channel operation for 0% duty cycle. Note that the A1 match posedge signal occurs at the same time as the B1=0x8 negedge signal. In this case A1 match has precedence over B1 match, causing the output pin to remain at EDPOL bit value, thus generating a 0% duty cycle signal.

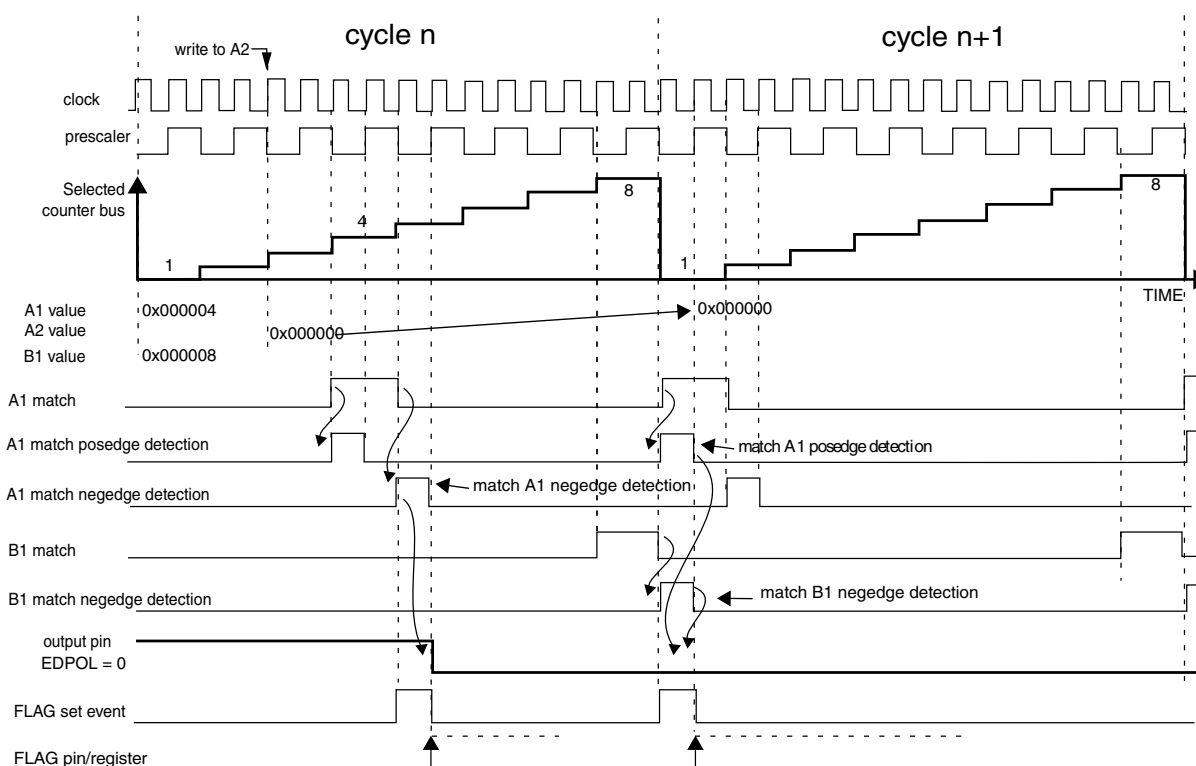


**Figure 34-153. OPWMB Mode with 0% Duty Cycle**

Figure 34-154 describes the operation of the OPWMB mode with the Output Disable signal being asserted. The Output Disable forces a transition in the output pin to the EDPOL bit value. After deasserted, the output disable allows the output pin to transition at the following A1 or B1 match. Note that the Output Disable does not modify the Flag bit behavior. Note that there is one system clock delay between the assertion of the output disable signal and the transition of the output pin to EDPOL.

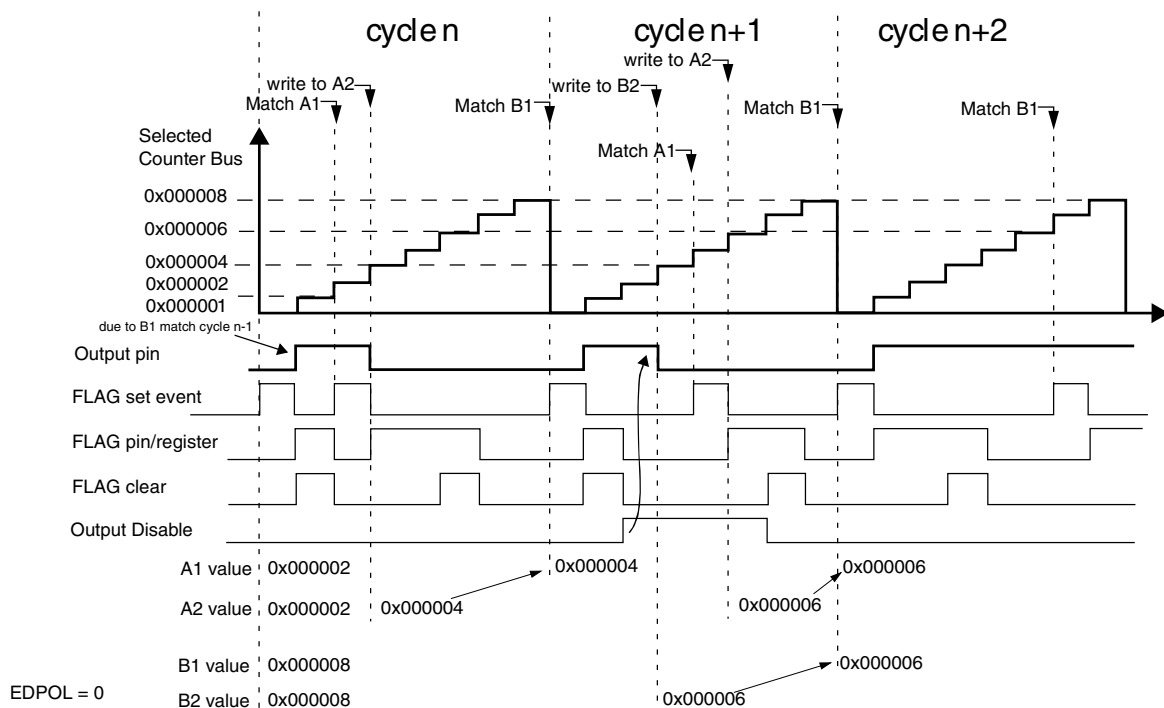


**Figure 34-154. OPWMB Mode with Active Output Disable (MODE[6] bit is 1b)**

Figure 34-155 shows a waveform changing from 100% to 0% duty cycle. EDPOL in this case is zero. In this example, B1 is programmed to the same value as the period of the external selected time base.

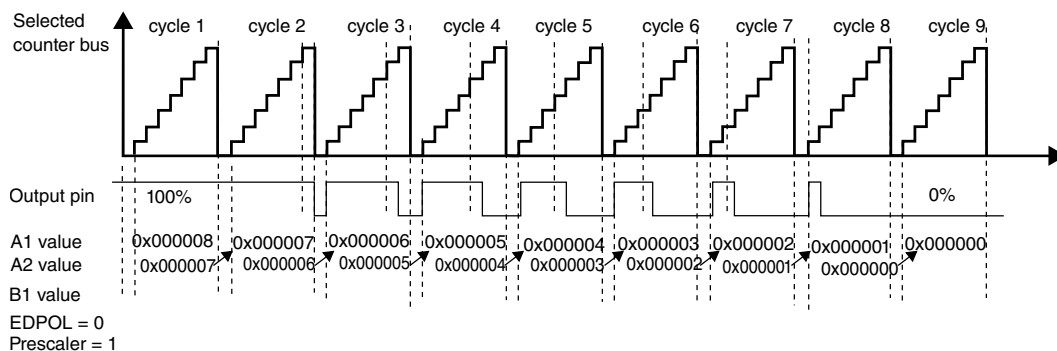


**Figure 34-155. OPWMB Mode from 100% to 0% Duty Cycle**

In Figure 34-155, if B1 is set to a value lower than 0x8, it is not possible to achieve 0% duty cycle by only changing A1 register value. Since B1 matches have precedence over A1 matches the output pin transitions to the opposite of EDPOL bit at B1 match. Note also that if B1 is set to 0x9, for instance, B1 match does not occur, thus a 0% duty cycle signal is generated.

### 34.6.1.2 Input Programmable Filter (IPF)

The IPF ensures that only valid input pin transitions are received by the Unified Channel edge detector. A block diagram of the IPF is shown in [Figure 34-156](#).

The IPF is a 5-bit programmable up counter that is incremented by the selected clock source, according to bits IF[0:3] in EMIOSC[n] register.

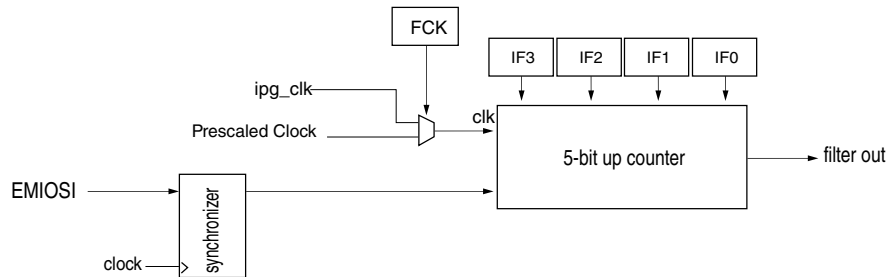


**Figure 34-156. Input Programmable Filter Submodule Diagram**

The input signal is synchronized by system clock. When a state change occurs in this signal, the 5-bit counter starts counting up. As long as the new state is stable on the pin, the counter remains incrementing. If a counter overflow occurs, the new pin value is validated. In this case, it is transmitted as a pulse edge to the edge detector. If the opposite edge appears on the pin before validation (overflow), the counter is reset. At the next pin transition, the counter starts counting again. Any pulse that is shorter than a full range of the masked counter is regarded as a glitch and it is not passed on to the edge detector. A timing diagram of the input filter is shown in [Figure 34-157](#).

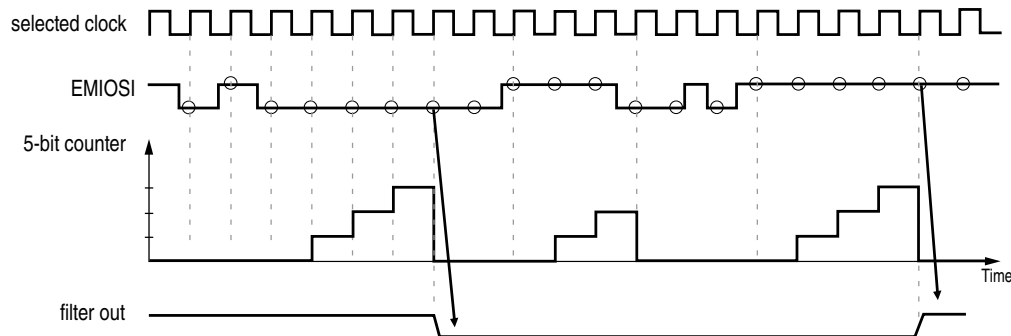


**Figure 34-157. Input Programmable Filter Example (IF[0:3] is 0010b)**

The filter is not disabled during either freeze state or negated GTBE input

### 34.6.1.3 Clock Prescaler (CP)

The CP divides the GCP output signal to generate a clock enable for the internal counter of the Unified Channels. The GCP output signal is prescaled by the value defined in the UCPRE[0:1] field description in the EMIOSC[n] register. The prescaler is enabled by setting the UCPREN bit in the EMIOSC[n] and can be stopped at any time by clearing this bit, thereby stopping the internal counter in the Unified Channel.

In order to ensure safe working and avoid glitches the following steps must be performed whenever any update in the prescaling rate is desired:

1. Write 0 at both GPREN bit in EMIOSMCR register and UCPREN bit in EMIOSC[n] register, thus disabling prescalers;
2. Write the desired value for prescaling rate at UCPRE[0:1] bits in EMIOSC[n] register;
3. Enable channel prescaler by writing 1 at UCPREN bit in EMIOSC[n] register;
4. Enable global prescaler by writing 1 at GPREN bit in EMIOSMCR register.

The prescaler is not disabled during either freeze state or negated GTBE input.

### 34.6.1.4 Effect of Freeze on the Unified Channel

When in debug mode, FRZ bit in the EMIOSMCR register and the FREN bit in the EMIOSC[n] are both set, the internal counter and Unified Channel capture and compare functions are halted. The UC is frozen in its current state.

During freeze, all registers are accessible. When the Unified Channel is operating in an output mode, the force match functions remain available, allowing the software to force the output to the desired level.

Note that for input modes, any input events that may occur while the channel is frozen are ignored.

When exiting debug mode or freeze enable bit is cleared (FRZ in the EMIOSMCR or FREN in the EMIOSC[n] register) the channel actions resume, but may be inconsistent until channel enters GPIO mode again.

## 34.6.2 IP Bus Interface Unit (BIU)

The BIU provides the interface between the Internal Interface Bus (IIB) and the Peripheral Bus, allowing communication among all submodules and this IP interface.



The BIU allows 8, 16 and 32 bits access. They are performed over a 32-bit data bus in a single cycle clock.

### 34.6.2.1 Effect of Freeze on the BIU

When the FRZ bit in the EMIOSMCR register is set and the module is in debug mode, the operation of BIU is not affected.

### 34.6.3 STAC Bus Client submodule

The STAC bus client provides one external time base, imported from the STAC bus, to the Unified Channels. [Figure 34-158](#) provides a block diagram for the module.

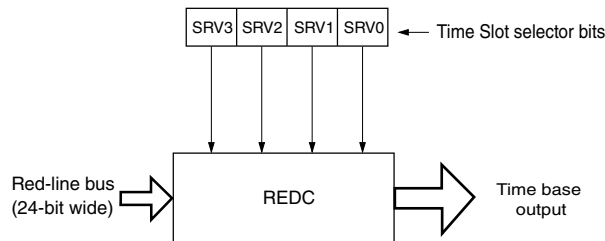
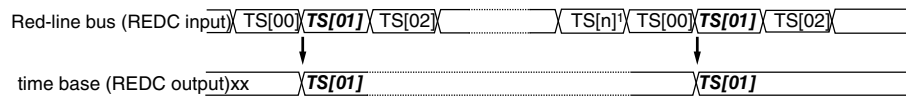


**Figure 34-158. Block diagram**

Bits SRV[0:3] in register EMIOSMCR selects the desired time slot of the STAC bus to be output. [Figure 34-159](#) shows a timing diagram for the REDC.



NOTES: 1. Maximum of 16 Time Slots (TS[n])  
2. In this case, SRV bits were set to capture TS[01]

**Figure 34-159. Timing diagram for the STAC bus and STAC output**

Every time the selected time slot change, the STAC output is updated.

See the chip-specific eMIOS information for the STAC server IDs.

### 34.6.3.1 Effect of Freeze on the STAC bus

When the FRZ bit in the EMIOS\_MCR register is set and the module is in debug mode, the operation of REDC submodule is not affected, i.e., there is no freeze function in this submodule.

### 34.6.4 Global Clock Prescaler Submodule (GCP)

The GCP divides the system clock to generate a clock for the CPs of the channels. The main clock signal is prescaled by the value defined in the Global Prescaler clock divider table within the eMIOS Module Configuration Register (EMIOSMCR) description according to the GPRE[0:7] bits in EMIOSMCR register. The global prescaler is enabled by setting the GPREN bit in the EMIOSMCR register and can be stopped at any time by clearing this bit, thereby stopping the internal counters in all the channels.

In order to ensure safe working and avoid glitches the following steps must be performed whenever any update in the prescaling rate is desired:

1. Write 0 at GPREN bit in EMIOSMCR register, thus disabling global prescaler;
2. Write the desired value for prescaling rate at GPRE[0:7] bits in EMIOSMCR register;
3. Enable global prescaler by writing 1 at GPREN bit in EMIOSMCR register.

The prescaler is not disabled during either freeze state or negated GTBE input.

#### 34.6.4.1 Effect of Freeze on the GCP

When the FRZ bit in the EMIOSMCR register is set and the module is in debug mode, the operation of GCP submodule is not affected, i.e., there is no freeze function in this submodule.

## 34.7 Initialization/Application Information

On resetting the eMIOS, the Unified Channels enter GPIO input mode.

### 34.7.1 Considerations

Before changing an operating mode, the UC must be programmed to GPIO mode and EMIOSA[n] and EMIOSB[n] registers must be updated with the correct values for the next operating mode. Then the EMIOSC[n] register can be written with the new operating mode. If a UC is changed from one mode to another without performing this procedure, the first operation cycle of the selected time base can be random, i.e., matches

can occur in random time if the contents of EMIOSA[n] or EMIOSB[n] were not updated with the correct value before the time base matches the previous contents of EMIOSA[n] or EMIOSB[n].

When interrupts are enabled, the software must clear the FLAG bits before exiting the interrupt service routine.

## 34.7.2 Application Information

Correlated output signals can be generated by all output operation modes. Bits OU[n] of eMIOS\_OUDIS register can be used to control the update of these output signals.

In order to guarantee that the internal counters of correlated channels are incremented in the same clock cycle, the internal prescalers must be set up before enabling the global prescaler. If the internal prescalers are set after enabling the global prescaler, the internal counters may increment in the same ratio, but at a different clock cycle.

It is recommended to drive Output Disable Input signals with the emios\_flag\_out signals of some UCs running in SAIC mode. When an output disable condition happens, the software interrupt routine must service the output channels before servicing the channels running SAIC. This procedure avoid glitches in the output pins.

### 34.7.2.1 Time Base Generation

For MC and OPWFM with internal clock source operation modes, the internal counter rate can be modified by configuring the clock prescaler ratio. [Figure 34-160](#) shows an example of a time base with prescaler ratio equal to one.

#### Note

MCB and OPWFMB modes have a different behavior.

PRE SCALED CLOCK RATIO = 1 (bypassed)

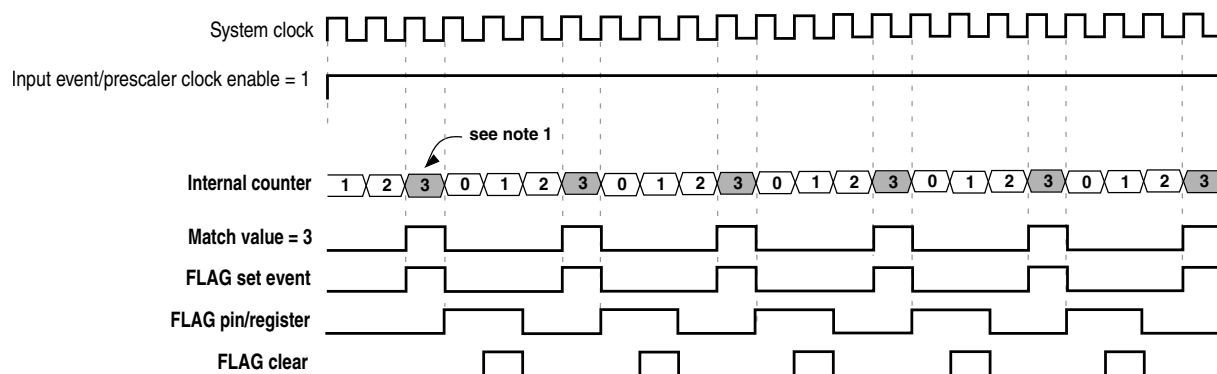


**Figure 34-160. Time base period when running in the fastest prescaler ratio**

If the prescaler ratio is greater than one or external clock is selected, the counter may behave in one of four different ways depending on the channel mode:

- If MC mode and Clear on Match Start and External Clock source are selected the internal counter behaves as described in [Figure 34-161](#).
- If MC mode and Clear on Match Start and Internal Clock source are selected the internal counter behaves as described in [Figure 34-162](#).
- If MC mode and Clear on Match End are selected the internal counter behaves as described in [Figure 34-163](#).
- If OPWFM mode is selected the internal counter behaves as described in [Figure 34-162](#). The internal counter clears at the start of the match signal, skips the next prescaled clock edge and then increments in the subsequent prescaled clock edge.

## Note

MCB and OPWFMB modes have a different behavior.

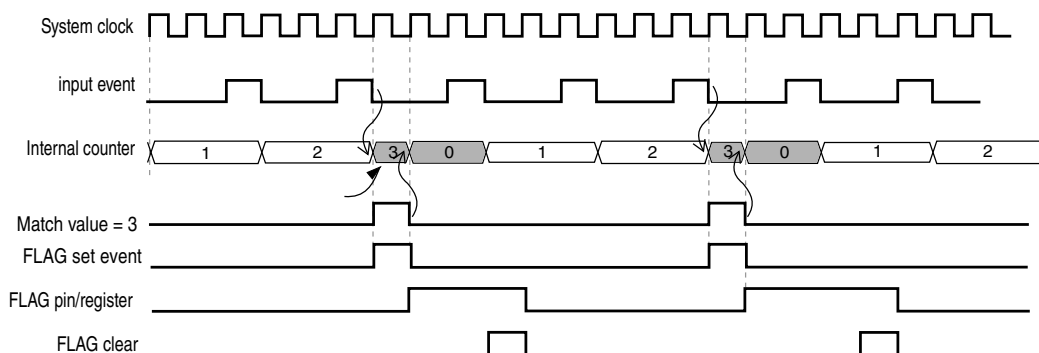
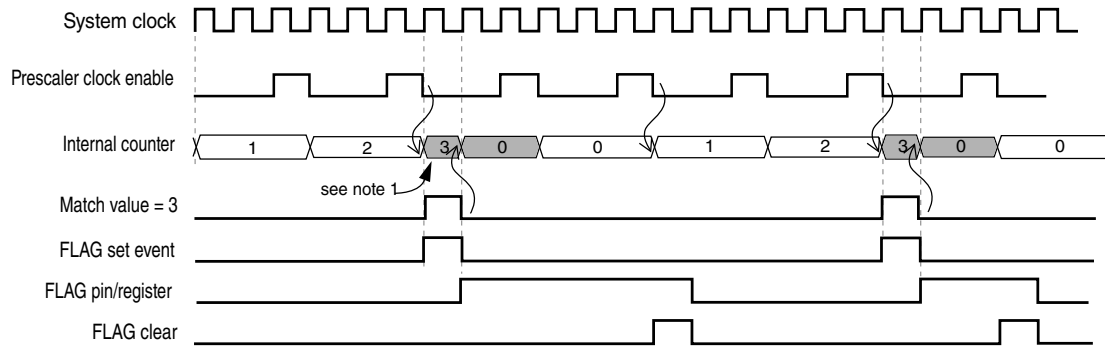


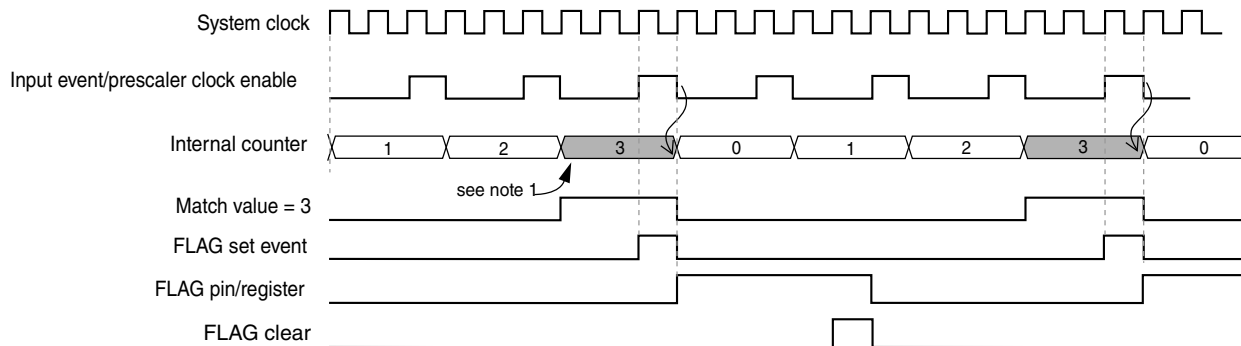
**Figure 34-161. Time base generation with external clock and clear on match start**

PRESCALED CLOCK RATIO = 3



Note 1: When a match occurs, the first clock cycle is used to clear the internal counter, and only after a second edge of pre scaled clock the counter will start counting.

**Figure 34-162. Time base generation with internal clock and clear on match start**



Note 1: The match occurs only when the input event/prescaler clock enable is active. Then, the internal counter is immediately cleared.

**Figure 34-163. Time base generation with clear on match end**

### 34.7.2.2 Coherent Accesses

Reading the EMIOSA[n] register again in the same period of the last read of EMIOSB[n] register may lead to incoherent results. This will occur if the last read of EMIOSB[n] register occurred after a disabled B2 to B1 transfer.

### 34.7.2.3 Channel/Modes Initialization

The following basic steps summarize basic output mode startup, assuming the channels are initially in GPIO mode:

1. *[global]* Disable Global Prescaler.
2. *[timebase channel]* Disable Channel Prescaler.

3. [*timebase channel*] Write initial value at internal counter.
4. [*timebase channel*] Set A/B register.
5. [*timebase channel*] Set channel to MC(B) Up mode.
6. [*timebase channel*] Set prescaler ratio.
7. [*timebase channel*] Enable Channel Prescaler.
8. [*output channel*] Disable Channel Prescaler.
9. [*output channel*] Set A/B register.
10. [*output channel*] Select timebase input through BSL[1:0] bits.
11. [*output channel*] Enter output mode.
12. [*output channel*] Set prescaler ratio (same ratio as timebase channel).
13. [*output channel*] Enable Channel Prescaler.
14. [*global*] Enable Global Prescaler.
15. [*global*] Enable Global Time Base.

The timebase channel and the output channel may be the same for some applications such as in OPWFM(B) mode or whenever the output channel is intended to run the timebase itself.

At any time the flags can be configured.

# Chapter 35

## Enhanced Time Processing Unit (eTPU)

### 35.1 Chip-specific eTPU information

This chip contains three eTPU modules with the following configuration:

- 32 channels per eTPU module
- 36 KB code RAM total
- 9 KB parameter RAM total
- ETPU\_A and ETPU\_B share the same program and data (parameter RAM) space in a dual-engine configuration
  - Support for up to 200 MHz clock rate (provides 100 MHz instruction execution rate)
  - ECC support on parameter RAM and program space
  - Sizes of shared eTPU\_A and eTPU\_B memories:
    - SCM instruction memory address space: 24 KB code RAM
    - SDM data memory address space: 6 KB parameter RAM
  - No shared subsystem support and therefore no Shared Subsystem Integration registers (SSIRs)
- ETPU\_C has private program and parameter RAM space in a single-engine configuration
  - Support for up to 200 MHz clock rate (provides 100 MHz instruction execution rate)
  - ECC support on parameter RAM and program space
  - Sizes of eTPU\_C memories:
    - SCM instruction memory address space: 12 KB code RAM
    - SDM data memory address space: 3 KB parameter RAM

See also this information:

- [eMIOS global time base synchronization with eTPU](#)
- [DSPI connections with SIU, eTPU, and eMIOS modules](#)
- [eMIOS and eTPU ODIS connections](#)

### 35.1.1 eTPU register field reset values

The following table lists chip-specific reset values for register fields of the eTPU instances.

**Table 35-1. Chip-specific eTPU register field reset values**

| eTPU instance | Register field                 | Reset value | Notes                                                               |
|---------------|--------------------------------|-------------|---------------------------------------------------------------------|
| eTPU_A/B      | eTPU_SCR_MCR[SCMSIZE]          | 01011b      | The reset value of eTPU_SCR_MCR is 000B_0000h.                      |
| eTPU_A/B      | eTPU_TBR_STAC_ENG1[SERVER_ID1] | 0000b       | This is consistent with eTPU_A TCR1 in <a href="#">Table 35-2</a> . |
| eTPU_A/B      | eTPU_TBR_STAC_ENG1[SERVER_ID2] | 0010b       | This is consistent with eTPU_A TCR2 in <a href="#">Table 35-2</a> . |
| eTPU_A/B      | eTPU_TBR_STAC_ENG2[SERVER_ID1] | 0001b       | This is consistent with eTPU_B TCR1 in <a href="#">Table 35-2</a> . |
| eTPU_A/B      | eTPU_TBR_STAC_ENG2[SERVER_ID2] | 0011b       | This is consistent with eTPU_B TCR2 in <a href="#">Table 35-2</a> . |
| eTPU_C        | eTPU_SCR_MCR[SCMSIZE]          | 00101b      | The reset value of eTPU_SCR_MCR is 0005_0000h.                      |
| eTPU_C        | eTPU_TBR_STAC_ENG1[SERVER_ID1] | 0100b       | This is consistent with eTPU_C TCR1 in <a href="#">Table 35-2</a> . |
| eTPU_C        | eTPU_TBR_STAC_ENG1[SERVER_ID2] | 0101b       | This is consistent with eTPU_C TCR2 in <a href="#">Table 35-2</a> . |
| eTPU_C        | eTPU_TBR_STAC_ENG2[SERVER_ID1] | —           | eTPU_C does not have engine 2.                                      |
| eTPU_C        | eTPU_TBR_STAC_ENG2[SERVER_ID2] | —           | eTPU_C does not have engine 2.                                      |

### 35.1.2 Access to eTPU registers in chip-level Stop mode

On this chip, if you attempt to access eTPU registers when the chip is in Stop mode (as defined in [Modes of operation](#)), you will receive a bus error.

#### NOTE

This Stop mode corresponds to the chip-level Stop mode. It is different from the eTPU Module Disable mode, also defined in [Modes of operation](#).

### 35.1.3 STAC server IDs

On this chip, an eTPU module can export the TCR1 and TCR2 timebases on the STAC bus to an eMIOS or eQADC module. See [STAC interface](#) for more information.

An eTPU module is able to export a timebase to an eMIOS or eQADC module when the eTPU timebase period is equal to or greater than 6 times the eTPU clock frequency, regardless of whether the clock ratio between the eTPU and eMIOS or eQADC is 1:1 or 2:1.



STAC server IDs are defined in the following table.

No more than 4 STAC servers must be enabled at any time.

**Table 35-2. STAC server IDs**

| eTPU instance | Time base | Server ID |
|---------------|-----------|-----------|
| eTPU_A        | TCR1      | 0         |
| eTPU_A        | TCR2      | 2         |
| eTPU_B        | TCR1      | 1         |
| eTPU_B        | TCR2      | 3         |
| eTPU_C        | TCR1      | 4         |
| eTPU_C        | TCR2      | 5         |

### 35.1.4 eTPU inputs with Glitch Filters

See [IGF interaction with other modules](#) for connections between eTPU and Input Glitch Filters (IGFs).

### 35.1.5 eMIOS and eTPU signal passthrough from pins via IGFs

On this chip, the IGF modules mediate between some pins and the eMIOS and eTPU modules. After system reset, the IGF modules are in Configuration mode by default. This mode disables IGF output and prevents the signals from the pins from reaching the eMIOS and eTPU modules.

The signals affected by this situation appear in [IGF interaction with other modules](#).

To route an affected signal directly to an eMIOS or eTPU instance, program the applicable channel of the applicable IGF instance:

1. Ensure that IGF\_MCRn[MDIS] is 0 and that the IGF channel is disabled: IGF\_MCRn[FGEN] is 0.
2. Configure the rising and falling edges for bypass:
  - Rising edge: Write 00b to IGF\_MCRn[RFM].
  - Falling edge: Write 00b to IGF\_MCRn[FFM].
3. Enable the IGF channel: Write 1 to IGF\_MCRn[FGEN].

At the end of this sequence, the programmed filter enters Normal mode and passes through the affected signal.

### 35.1.6 eTPU output disable channel groups

The output disable channel groups are defined in the following table. See [Table 34-2](#) for additional information.

**Table 35-3. Output disable channel groups**

| Output disable signal | eTPU_A channels | eTPU_B channels | eTPU_C channels |
|-----------------------|-----------------|-----------------|-----------------|
| ODIS_1(0)             | 0 to 7          | –               | 0 to 7          |
| ODIS_1(1)             | 8 to 15         | –               | 8 to 15         |
| ODIS_1(2)             | 16 to 23        | –               | 16 to 23        |
| ODIS_1(3)             | 24 to 31        | –               | 24 to 31        |
| ODIS_2(0)             | –               | 0 to 7          | –               |
| ODIS_2(1)             | –               | 8 to 15         | –               |
| ODIS_2(2)             | –               | 16 to 23        | –               |
| ODIS_2(3)             | –               | 24 to 31        | –               |

### 35.1.7 Control of the eTPU Output Buffer Enable signal

On this chip, the eTPU Output Buffer Enable signal is controlled by the Pad Configuration Registers (PCRs) in the SIU. See [Table 8-3](#) and [Pad Configuration Register \(SIU\\_PCR<sub>n</sub>\)](#).

## 35.2 Device-specific Information

- Single (32 channels) or Dual (64 channels, dual core) Engine
- SCM size, error detection, error correction
- SDM size, error detection, error correction
- Nexus class 3 support
- GTBE is a logical OR with other modules (eMIOS, other eTPU)
- channel output buffers are controlled through the SIU, and cannot be controlled with microcode

- channels used to trigger the eQADC
- channels used to enable the Reaction Module modulation
- channels with DMA request
- TRC1, TRC2 STAC server IDs

## 35.3 Introduction

eTPU is an intelligent, semi-autonomous co-processor designed for timing control. Operating in parallel with the Host CPU, the eTPU processes instructions and real-time input events, performs output waveform generation, and accesses shared data without Host intervention. Consequently, for each timer event, the Host CPU setup and service times are minimized or eliminated.

High-level assembler, compiler and documentation allows customers to develop their own functions on the eTPU.

eTPU architecture aims at high resolution timing capabilities. From a system perspective, high resolution timing is limited by Host CPU overhead required for servicing timing tasks such as period measurement, pulse measurement, pulse width modulated waveform generation, etc. On the eTPU, high resolution timing is achieved by three main capabilities:

- Reduced latency: pin actions are immediate.
- Reduce or eliminate host interrupt service time.
- Double action channel capability reducing the channel request rate.

eTPU provides higher resolution than the Host CPU can achieve and creates no Host overhead for servicing timing tasks.

Latency is the interval from occurrence of an event to the start of event servicing. eTPU can service its own events without interrupting the Host. There are two types of timing events:

- Input pin transition.
- Selected Time Base match, i.e., a selected Time Base counter reached or exceeded a pre-programmed value.

Service time is the time spent servicing an event. In general, in microcontrollers the service time is constrained because the instruction set is not optimized for time function synthesis. The eTPU instruction set is optimized, so that time functions can be implemented with much fewer instructions than the Host CPU. Instructions execute faster, service time is reduced and program memory compacted.

Instructions executed by the eTPU are connected directly to the eTPU timing hardware and allow parallelism of hardware related actions.

### 35.3.1 Overview

The figure below shows a top-level eTPU block diagram. It displays a dual eTPU Engine configuration.

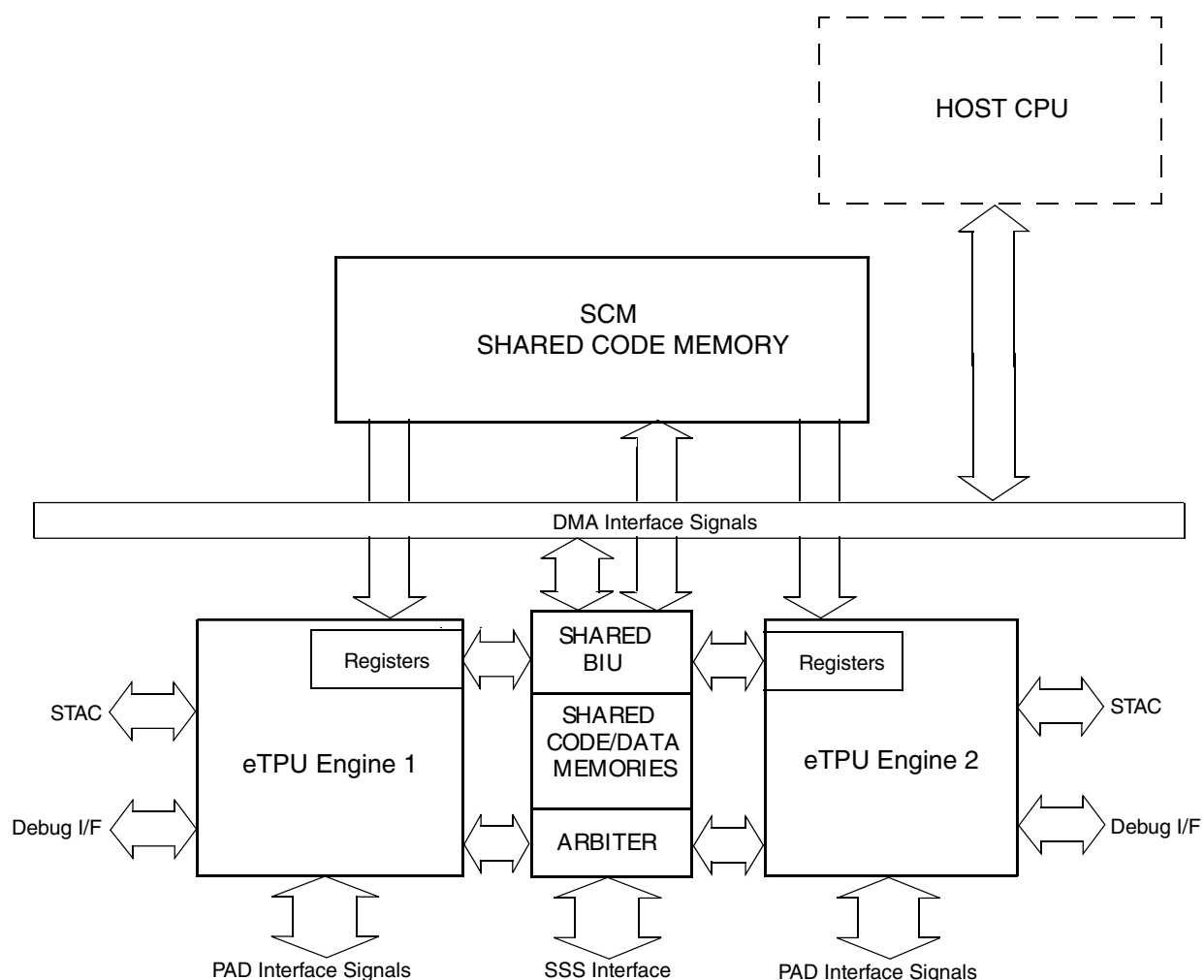


**Figure 35-1. eTPU Block Diagram**

**eTPU Engine** is responsible for processing input pin transitions and output pin waveform generation based on the **Time Bases**. Each eTPU Engine has its own microprocessor and dedicated hardware for processing signals on I/O pins and can also interface with external time bases through the STAC bus.

Both eTPU Engine CPUs, hereafter called **microengines**, fetch microinstructions from a **Shared Code Memory - SCM**.

**Shared Data Memory - SDM** - holds eTPU application parameters and work data. It is accessed by Host and both microengines.

**Bus Interface Unit - BIU** - allows Host to access eTPU registers, SCM and SDM.

Each I/O signal pair is associated with a dedicated **Channel**, which provides hardware for input signal processing and output signal generation, in relationship with selected Time Bases.

The eTPU, as a microprocessed subsystem, works much like a typical real-time system: it runs microengine code from instruction memory (SCM) to handle specific events, accessing data memory (SDM) for parameters, work data and application state info; events may originate from I/O Channels (due to pin transitions and/or time base matches), Host CPU requests or inter-channel requests; events that call for local eTPU processing activate the microengine by issuing a **Service Request**. The Service Request microcode may set an interrupt to the Host CPU. I/O channel events cannot directly interrupt the Host CPU.

Each Channel is associated with a **Function**, which defines its behavior: the Function is a software entity consisting, within the eTPU, of a set of microengine routines that attend to Service Requests. The Function routines are also responsible for Channel configuration. Function routines reside in SCM, which may contain several Functions. A Function may be assigned to several Channels, but a Channel can be associated with just one Function at a given moment. The association between Functions and Channels is defined by Host CPU, and is explained in detail in [Functions and threads](#).

eTPU hardware supplies resource sharing features that support concurrency:

- A hardware **Scheduler** dispatches the Service Request microengine routines based on a set of priorities defined by the Host CPU. Each Channel has its associated priority.
- A Service Request routine cannot be interrupted until it ends. This sequence of uninterrupted instruction execution is called a **Thread**.

- Channel-specific context (registers and flags) is automatically switched between the end of a Thread and the beginning of the next one.
- SDM arbitration, a dual-parameter coherency controller and semaphores can be used to ensure coherent access to eTPU data shared by both eTPU Engines and Host CPU.

### 35.3.1.1 eTPU Engine

Each eTPU Engine consists of all blocks that are duplicated in a dual eTPU configuration: two 24-bit time bases, 32 independent timer channels, a task scheduler, a microengine, and a Host interface. In addition, a 32-bit Shared Data Memory (SDM) is used for two eTPU Engines data storage and for passing information between the eTPU Engines and the Host CPU.

The following figure shows the block diagram for the eTPU Engine.

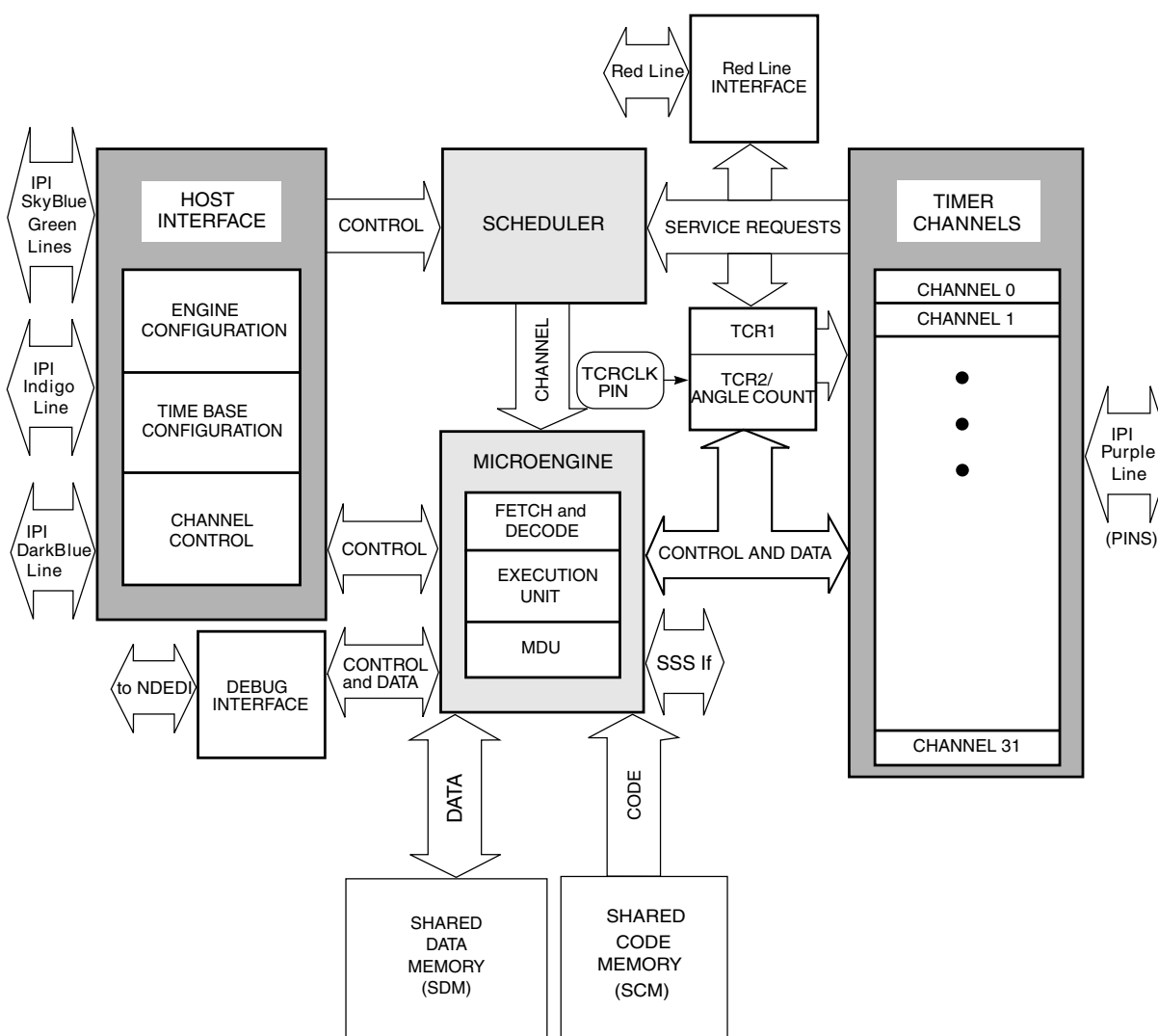


**Figure 35-2. eTPU Engine Block Diagram**

eTPU Engines 1 and 2 are sometimes called eTPU 1 and eTPU 2 for short throughout this chapter.

### 35.3.1.1.1 Time Bases

Two 24-bit counters TCR1 and TCR2 provide reference time bases for all match and input capture events. Prescalers for both time bases are controlled by the Host CPU through bit fields in the eTPU Engine configuration registers. The eTPU is able to export/import time to/from TCR1 or TCR2 in accordance to the STAC bus specification.

The clock for each of TCR1 and TCR2 clock can be independently derived from the eTPU clock or from an external input via the TCRCLK clock pin. In addition, the TCR2 timebase can be derived from special angle-clock hardware which enables implementing angle-based functions. This feature is added to support advanced angle based engine control applications.

For further details refer to [Time bases](#).

### 35.3.1.1.2 eTPU Timer Channels

Each eTPU Engine has 32 independent channels, each corresponding to an Input/Output signal pair. The channel time resolution is 24 bits, and are all identical.

Each channel consists of logic which supports two events and output controls. The event logic contains two 24-bit capture registers, two 24-bit match registers, greater-equal and equal-only comparators. Supporting two events enables many combinations of double action functions (for example the channel can handle two events with a single microcode service).

The channel configuration can be changed immediately by the microengine. Each channel can perform double capture, double match and other capture-match combinations. Channel modes available can do ordered or unordered match. Some modes are also provided that can block one match by the occurrence of the other. Service request can be generated on one or both of the match events.

Input signal can be separated from output signal in each channel. They can, optionally, be combined in a single I/O pin driver. An output buffer enable is controlled by the SIU (or SIUL). Digital filters are provided for the input signals, with distinct filtering modes available.

Each channel can use any time base or angle counter for either match or capture operation. For example, a match on TCR1 can capture the value of TCR2. The channels can request service from the microengine due to recognized pin transitions (input events) or timebase matches.

The eTPU channels also support the basic single-action operations found on TPU3 functionality with the exception that time resolution is 24-bits.

Channel configuration combinations:

- Single input capture, no match (TPU3 functionality)
- Single input capture with single match timeout (TPU3 functionality)
- Single input capture with double match timeout with several double match sub-modes
- Double input capture with single or double match timeout with several double match sub-modes
- Single output match (TPU3 functionality)



- Double output match with several double match sub-modes
- Input-dependent output generation

The double match functionality has various combinations for generation of service request and determining pin actions. For more details refer to [Enhanced Channels](#).

In addition to the predefined channel configurations above, the user can also program its own channel configuration, defining how input captures, matches and service-requests are related.

### 35.3.1.1.3 Host interface

The Host interface allows the Host CPU to control the operation of the eTPU. The Host CPU must initialize the eTPU by writing to the appropriate Host interface registers to assign a Function and priority to each channel. In addition, the Host writes to the Host Service Request and channel configuration registers to further define Function operation for each initialized channel. Refer to [Host interface](#) for a detailed description.

When the SCM is implemented by RAM, the Host must first initialize it with the proper microcode program prior to enabling any eTPU Function, and then enable eTPU access (which also disables Host access).

### 35.3.1.1.4 Shared Data Memory - SDM

The SDM works as data RAM which can be accessed by the Host CPU and up to two eTPU Engines. This memory is used for information transfer between the Host CPU and the eTPU, as data storage for the eTPU microcode program or for communication between the two eTPU Engines. SDM width is 32 bits, and is accessible by the Host as byte, 16-bit or 32-bit wide. eTPU can access it as full 32-bits, lower 24-bits or upper byte (8-bit).

The host can also access the SDM space mirrored in other area with Parameter Sign Extension (PSE). Parameter Sign Extension accesses differ from the usual host accesses to the original SDM area as follows:

- Writes are effective only to the lower 3 bytes of a word: the word's most significant byte is kept unaltered in SDM.
- Reads return the lower 3 bytes of a word sign-extended to 32 bits, i.e.: the most significant bit of the word's 2nd most significant byte is copied in all 8 bits of the most significant read byte.

Each eTPU channel can be associated with a variable number of parameters located in the SDM, according to its selected Function. In addition, the SDM can be fully shared between two eTPU Engines, enabling direct communication between them.

High flexibility of the SDM utilization is achieved as follows:

- Each channel has a programmable base address pointing to the address of its first parameter with two parameter granularity. This way the SDM can be partitioned according to the actual function needs.
- The microcode can access the first 128 parameters of the selected channel in channel relative access mode.
- Each Engine can access all the SDM address space in indirect addressing mode. Blocks of data are easily transferred using stack operation.
- Absolute addressing mode can access the first 256 parameters (TPU3 functionality), implementing a shared pool of parameters holding global variables.

In the Host address space each parameter occupies four bytes. eTPU usage of the upper byte is achieved by having a 32-bit P register which can access the upper byte, the lower 24 bits or all the 32 bits. The microcode can switch between access sizes at any time.

Each Function may require a different number of parameters. During the eTPU initialization the Host has to program channel base addresses, allocating proper parameters for each channel according to its selected Function.

#### **35.3.1.1.5 Shared Subsystem Area**

The Shared Subsystem (SSS) area is a memory map region used by both the host and the eTPU to access registers of the reaction channel, giving the user the option to control them by the host, the eTPU code, or a combination of both.

#### **35.3.1.1.6 Scheduler**

Out of reset, all channels are disabled. The Host CPU makes a channel active by assigning it one of three priorities: high, middle, or low. The Scheduler determines the order in which channels are serviced based on channel number and assigned priority. The priority mechanism, implemented in hardware, ensures that all requesting channels are serviced. For additional details refer to [Scheduler](#).

### 35.3.1.1.7 Microengine

eTPU microengine is a simple RISC implementation which performs each instruction in a microcycle of two eTPU clocks, while prefetching the next instruction through an instruction pipeline. Instruction execution time is constant unless it gets wait states from the SDM arbitration. Two eTPU Engines share code memory without having any performance degradation by interleaving their accesses (the Shared Code Memory has one-clock access time).

Instruction width is 32 bits. The microengine instruction set provides basic arithmetic and logic operations, flow control (jumps and subroutine calls), SDM access, and Channel configuration and control. The instruction formats are defined in such a way that allow particular combinations of two or three of these operations with unconflicting resources to be executed in parallel in the same microcycle.

Microengine has also an independent Multiply/Divide/MAC unit that performs these complex operations in parallel with other microengine instructions.

Channel functionality is tightly integrated to the instruction set through Channel Control operations and conditional Branch operations, which support jumps/calls on Channel-specific conditions. This allows quick and terse Channel configuration and control code, contributing to reduced service time.

Detailed description can be found in [Microengine](#).

### 35.3.1.1.8 Dual eTPU Engine System

A typical eTPU implementation includes two eTPU Engines sharing SDM and the same code in SCM.

The two eTPU Engines share the Bus Interface Unit (BIU) and the data memory (SDM) which enable Host-eTPU and eTPU Engine-Engine communication. The shared BIU includes coherency logic which supports dual parameter (8 bytes) coherency in transfers between Host and eTPU, using a temporary parameter area within the SDM. More details on this can be found on [Parameter sharing and coherency](#).

## 35.3.2 Features

The following sections show eTPU features and enhancements.

### 35.3.2.1 eTPU Feature Summary

The eTPU includes these distinctive features:

- up to 32 channels for each eTPU Engine, each channel is associated with an Input/Output signal pair.
  - enhanced input digital filters on the input pins for improved noise immunity. The eTPU digital filter can use 2 samples, 3 samples or work in continuous mode.
  - identical, orthogonal channels, except for channel 0: each channel can perform any time function. Each time function can be assigned to more than one channel at a given time, so each signal can have any functionality. Channel 0 has the same capabilities of the others, but can also work with special Angle Counter logic (see below).
  - Link Service Request allows activation of a Channel function by request of another channel, even between eTPU Engines.
  - Host Service Request allows activation of a Channel function by Host CPU request
  - each channel has an event mechanism which supports single and double action functionality in various combinations. It includes two 24-bit capture registers, two 24-bit match registers, 24-bit greater-equal and equal-only comparators.
- Two independent 24-bit time bases for channel synchronization:
  - first time base clocked by eTPU clock with programmable prescaler division from 1 to 512 (in steps of 2), or by output of second time base prescaler.
  - first time base can also be clocked by external signal with programmable prescaler division of 1 to 256.
  - second time base clocked by external signal with programmable prescaler division from 1 to 64.
  - second time base external clock source can be replaced by eTPU clock divided by 8.
  - both time bases can be exported or imported via Shared Time and Counter bus.
  - second time base counter can work as an Angle counter, enabling angle based applications to match angle instead of time.
  - second time base can also be used as a pulse accumulator gated by external signal.
- Event-Triggered RISC processor (microengine):
  - 3 stage pipeline implementation (fetch, decoding and execution), with separate instruction memory - SCM - and data memory - SDM (Harvard architecture)

- fixed-length instruction execution in one microcycle of two eTPU clocks
- interleaved SCM access in dual eTPU Engine avoids contention in time for instruction memory
- SCM address space of up to 16K positions (64 Kbytes) (see chip-specific eTPU information)
- SDM with interleaved access in dual eTPU Engine avoids contention for data memory
- SDM address space (both Engines) of up to 8 Kbytes (see chip-specific eTPU information)
- instruction set with embedded Channel support, including specialized Channel control subinstructions and conditional branching on Channel-specific flags.
- channel-oriented addressing: channel-bound address mode with Host configured Channel Base Address allows channel data isolation, independent of microengine application code.
- channel-bound data address space of up to 128 32-bit parameters (512 bytes)
- global parameter address mode allows access to common Channel data of up to 256 32-bit parameters (1024 bytes)
- support for indirect and stacked data access schemes.
- parallel execution of: data access, ALU, Channel control and flow control subinstructions in selected combinations.
- 32-bit microengine registers and 24-bit resolution ALU, with 1 microcycle addition and subtraction, absolute value, bitwise logical operations on 24-bit, 16-bit, or byte operands; single bit manipulation, shift and rotate operations, sign extension and conditional execution.
- additional 24-bit Multiply/MAC/Divide unit which supports all signed/unsigned Multiply/MAC combinations, and unsigned 24-bit Divide. The MAC/Divide unit works in parallel with the regular microcode commands.
- Resource sharing features support channel sharing of channel registers, memory and microengine time:
  - hardware Scheduler works as a "task management" unit, dispatching event service routines by predefined, Host-configured priority.

- automatic Channel context switch when a "task switch" occurs, i.e., one Function Thread ends and another begins to service a request from other Channel: Channel-specific registers, flags and parameter base address are automatically loaded for the next serviced channel.
- individual channel priority setting in 3 levels: high, middle and low.
- Scheduler priority scheme allows calculation of worst case latency for event servicing and ensures servicing all channels by preventing permanent blockage.
- SDM shared between Host CPU and both eTPU Engines, supporting communication either between Channels and Host or inter-channel.
- hardware implementation of 4 Semaphores supports resource sharing between both eTPU Engines.
- Hardware semaphores directly supported by the microengine instruction set.
- dual parameter coherency hardware support allows atomic (to host) access to 2 parameters by microengine(s) in back-to-back accesses.
- coherent dual-parameter controller allows atomic (to microengines) accesses to 2 parameters by the host.
- Development support features:
  - Nexus class 3 debug support (optional, associated with the eTPU-Nexus Block NDEDI).
  - Software breakpoints.
  - Debug interface supporting single-step execution, forced microinstruction execution, Hardware breakpoints and watchpoints on several conditions.
- Safety support features:
  - SCM (code memory) continuous signature-check built-in self test (MISC - Multiple Input Signature Calculator), runs concurrently with eTPU normal operation.
  - See [eTPU Plus Enhancements over eTPU](#) for additional safety features.

### 35.3.2.2 eTPU Enhancements over TPU3

- 32 orthogonal channels with enhanced functionality. Full support for double action with double match and double transition sub-mode combinations.

- Input and Output features separated in channel logic and microinstructions, allowing input and output signals to be processed separately or combined.
- Increased time resolution and execution unit to 24 bits.
- Increased linear code memory, shared by two eTPU Engines, configurable up to 16K positions (64 KB).
- Increased SDM address range (8 KB each Engine) and width (32 bits per parameter). The SDM can be dynamically allocated to support variable number of parameters for each channel. Each channel can have access to at least 256 parameters.
- The SDM is fully shared by two eTPU Engines (SDM), supporting direct inter-engine communication with the help of hardware semaphores.
- Enhanced arithmetic operations, including add/subtract with carry, absolute value, multiple shift and rotate, conditional execution with variable operand widths.
- Enhanced logic operations, including bitwise operations (and, or, xor) and bit manipulation, with conditional execution. Support for read-modify-write of any bit in the SDM.
- Hardware for Multiply/MAC/Divide, running in parallel to execution of other operations. The 24-bit divide result is available after 13 other unrelated instructions. Multiplication supports any data width of both operands (8, 16 or 24 bits), signed or unsigned. A 24x24 Multiply/MAC result is available after four other unrelated instructions. A 24x8 Multiply/MAC result is available after one other unrelated instruction.
- Supports export/import of time bases from other sources through the real time bus (STAC - Shared Time and Counter bus). This internal bus is used for sharing real time data between multiple peripherals.
- Contains angle clock hardware, supported by microcode, which can provide a 24-bit angle bus instead of time bus. This feature enables the eTPU to run angle based engine control applications.
- More interrupt types. Each eTPU channel can generate a data transfer request interrupt, in addition to regular interrupts, and one global exception interrupt. Data Transfer requests can be used either as interrupt sources or DMA requests. This feature takes advantage of DMA peripherals which off-load the Host. Interrupt Overflow status is also provided.
- Improved visibility to the Host (pin states, time bases, serviced channel).

- An edge case of priority inversion on TPU3 Scheduler was resolved.
- Supports channel link requests between eTPU Engines.

### 35.3.2.3 eTPU Plus Enhancements over eTPU

eTPU Plus features include the enhancements below in addition to all eTPU features

- TCR1, channel logic and digital filters (both channel and TCRCLK) now have an option to run at divisions of full eTPU clock speed, besides eTPU clock / 2 (see [T2/T4 channel timing](#)).
- Channels support unordered transitions: transition B can now be detected before transition A. Related to this enhancement, TDLA and TDLB can now be independently negated by microcode (see [Channel Mode Logic and Event Flags](#))
- Channel logic can now work on full eTPU clock, allowing faster response with slow clocks (see [T2/T4 channel timing](#)).
- Added a new User Programmable Channel Mode: the blocking, enabling, service request and capture characteristics of this channel mode can be programmed via microcode (see [General Channel registers](#)).
- Microinstructions now provide an option to issue Interrupt and Data Transfer requests selected by CHAN. They can also be requested simultaneously at the same instruction (see [Channel Interrupt and Data Transfer Requests](#)).
- Channel Flags 0 and 1 can now be tested for branching, besides selecting the entry point (see [Branch Operations](#)).
- Channel digital filters can be bypassed, and can be clocked by full eTPU clock (see [Bypass Mode](#)).

Backward compatibility is maintained so that all legacy object code (both Host and microcode) runs on eTPU Plus without modification.

### 35.3.2.4 eTPU2 enhancements over eTPU Plus

eTPU2 features include the enhancements below in addition to all eTPU Plus enhancements and eTPU features.



- Scheduler priority-passing mechanism can now be disabled (see [Priority passing disabling](#)).
- New Watchdog mechanism kills threads over a programmable timeout (see [Watchdog](#)).
- New counter allows microengine load information collection for performance analysis (see [Idle Counter](#)).
- Channels 1 and 2 (besides channel 0) can now be selected to control the EAC (see the TBCR\_ENG1 and TBCR\_ENG2 eTPU Time Base Configuration Registers).
- Timebase prescalers are now reset when the GTBE input is negated, guaranteeing synchronization with eMIOS in all cases (see [GTBE - Global Time Base Enable](#)).
- New MISC flag indicates when an SCM signature calculation round is completed. This allows measuring of the average MISC scan period in a real application situation (see the eTPU Module Configuration Register - MCR).
- New channel TCCEA flag allows continuous capture even after TDLA is set, making it fully compatible with TPU behavior (see [TCCEA - Transition Continuous Capture Enable](#)).
- New branch condition PRSS tells the pin state at the time when a channel (match or transition) service request occurred (see [Pin Control Registers](#)).
- MRLEA/B can now be negated independently by microcode (see [Disable Matches](#)).
- New Engine Relative address mode allows a function to access SDM address space common to one engine, but distinct between engines (see Engine Relative Addressing Mode under [SDM Addressing Modes](#)).
- Memory error support for Code (SCM) and Data (SDM) memories: error detection, correction, and hardware-assisted soft-error fixing (see [Memory error support](#)).
- Sharing of microengine and host accesses to subsystem modules (for instance, the Reaction Module), allowing their control by the eTPU, the host, or both (see [Shared Subsystems support](#)).

Backward compatibility is maintained so legacy object code (both Host and microcode) runs on eTPU2 without modification.

### 35.3.2.5 eTPU2 Plus enhancements over eTPU2

eTPU2 Plus features include the enhancements listed below in addition to all eTPU features, eTPU Plus enhancements and eTPU2 enhancements.

- New Latency Monitor allows latency measurements, either for individual channels or channels combined in a single priority (High, Medium or Low).
- Watchdog can now be configured to disable thread forced ending and channel blocking, allowing it to monitor thread lengths with no interference on the application.

- EAC Angle Tick Generator can now be clocked by the TCR2 prescaler output, as an alternative to the TCR1 clock.
- EAC now supports up to 7 missing teeth on hardware.

Backward compatibility is maintained so legacy object code (both Host and microcode) runs on eTPU2 Plus without modification.

### 35.3.3 Modes of operation

eTPU can be seen as capable of working in the following modes:

- **User Configuration Mode**

User has the ability to program the eTPU Cores with User Time Functions, having access to the Shared Code Memory (SCM).

- **User Mode**

User does not access the eTPU Shared Code Memory:

- Use of predefined eTPU Functions
- No need for eTPU Core programming ability

- **Debug Mode**

User debugs eTPU code, accessing special Trace/Debug features via Nexus interface:

- hardware breakpoint/watchpoint setting
- access to internal registers
- single-step execution
- forced instruction execution
- software breakpoint insertion and removal.

- **Module Disable Mode**

eTPU Engine clocks are stopped through a register write to the MDIS field in [eTPU Engine 1 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG1\)](#) and [eTPU Engine 2 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG2\)](#), saving power. Input sampling stops. eTPU Engines can be in Module Disable Mode independently. Module Disable Mode stops only the Engine clock, so that the Shared BIU, and Global

Channel registers can be accessed, and interrupts and DMAs can be cleared and enabled/disabled. An Engine only enters Module Disable Mode when any currently running thread is finished (see [Functions and threads](#)).

- **Stop Mode**

Stop Mode is entered when eTPU answers device stop request assertion with stop acknowledge. The definition of which clocks are stopped is made at the MCU level, which defines whether or not registers can be accessed, interrupts and DMA requests cleared.

These modes are loosely selected: there is no unique register field or signals to choose between them. Some features of one mode can be used with features of other mode(s). More on this subject can be found on [eTPU mode selection](#) below.

### Note

Throughout this chapter, an engine is said to be "stopped" if it is either in Module Disable mode or Stop mode.

### 35.3.3.1 eTPU mode selection

User and User Configuration are the production operating modes, and differ from each other only in access to SCM. User programmability is only possible with a RAM SCM.

On chips where the SCM is implemented as a RAM, it can either be accessed directly from IP-Bus for code loading, or for software breakpoint setting. On chips with a ROM SCM, an internal SCM Emulation RAM may be used, depending on the specific MCU implementation, to replace ROM SCM for test or debug purposes. SCM Emulation RAM is selected in a MCU-specific way. For more details, see [SCM Emulation](#).

Debug Mode is characterized by the use of the debug interface features. Specifically, Nexus blocks provide Nexus class 3 debug features. For more information on debug features, refer to [Development support features](#).

Module Disable Mode is entered by setting the MDIS field in [eTPU Engine 1 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG1\)](#) and [eTPU Engine 2 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG2\)](#). eTPU Engines can be individually stopped going into Module Disable Mode (there is one eTPU\_SCR\_ECR register for each Engine). Each engine can leave Module Disable Mode by writing MDIS=0 (which can only be done if VIS=0).

Stop Mode is activated by IP-Bus ( device stop request). In this case, the eTPU waits for both eTPU Engines to enter in stop mode, and then asserts the stop acknowledge line. eTPU leaves Stop Mode when device stop request is negated, but only if VIS=0. If device stop request is negated and VIS=1, eTPU will leave Stop Mode as soon as VIS=0.

### **Note**

An Engine can stay in Module Disable mode when it leaves Stop Mode if its bit MDIS=1, even if the other leaves it.

## **35.4 Detailed signal descriptions**

Detailed signal descriptions are presented in the following sections.

### **35.4.1 eTPU channel output signals**

Each channel output signal is associated with a channel. The microcode may affect the logic level of an output signal<sup>1</sup> by implementing one of two actions:

- Specify the logic level output to the signal when there is a match or a transition.
- Immediately force a logic level.

The output signal may also be forced to a logic level, independently of the output value from the channel logic, by one of the four (each Engine) output disable input signals (see [eTPU channel output disable signals](#)).

The output buffer enable signals are not used.

### **35.4.2 eTPU channel input signals**

Each channel input signal is associated with a channel. The microcode can directly control the effect of the transition edge. Each channel can be programmed to sense a transition when a rising and/or falling edge is detected. The channel logic can also process two transition events, and relate these events to each other and to other programmed timer events. The edge sensitivities of the two transition events are configured independently by microcode. For further information refer to [Enhanced Channels](#) and [Transition Detection and Pin Action Control](#).

---

1. Note that the minimum pulse width is one microcycle (two eTPU clocks), and slow pads may not be able to transfer it on time. For generation of very short pulses the eTPU pads have to be programmed by the system integration for fast operation mode with the voltage levels defined for fast pad operation in the MCU technology.

Each channel input signal has an associated synchronizer made of two flip-flops sampling the signal on every other eTPU clock<sup>2</sup>, followed by a digital filter. This digital filter can work in three sub-modes, whose purpose is to filter out noise pulses that have width less than a programmed value of eTPU clocks, preventing these transitions from being input to the transition detect logic. The synchronizer and digital filter are guaranteed to pass pulses that are greater than a programmed value. All channel input filters in one Engine work on the same mode and sampling clock. For more information on channel input filters, refer to [Enhanced Digital Filter - EDF](#). In one of the Angle Modes, the output of the digital filter of channel 0 is replaced by the output of TCRCLK signal digital filter (see [EAC - eTPU Angle Counter](#)).

### 35.4.3 Time Base Clock Signal — TCRCLK

TCRCLK is an input signal used to control the Time Bases TCR1 and TCR2. There is one independent TCRCLK input for each Engine. For pulse accumulator operations TCRCLK can be used as a gate for a counter based on the eTPU clock divided by eight. For Angle operations TCRCLK can be used to get the tooth transition indications in Angle Mode. Further details can be found on [Time bases](#) and [EAC - eTPU Angle Counter](#).

Like the channel input signals, the TCRCLK signal has an associated synchronizer followed by a digital filter. This digital filter can work in two sub-modes, whose purpose is to filter out noise pulses that have width less than a programmed value of eTPU clocks, preventing these transitions from being input to the transition detect logic. The synchronizer and digital filter are guaranteed to pass pulses that are greater than a programmed value. The clock and operation sub-mode of the TCRCLK filter is configured independently of the other channel input filters, through the TCRCF field in [eTPU Engine 1 Time Base Configuration Register \(eTPU\\_TBR\\_TBCR\\_ENG1\)](#) and [eTPU Engine 2 Time Base Configuration Register \(eTPU\\_TBR\\_TBCR\\_ENG2\)](#). For more information on filter sub-modes, see [TCRCLK digital filter](#). In one of the Angle Modes, the output of the digital filter of channel 0 is replaced with the output of TCRCLK signal digital filter (see [EAC - eTPU Angle Counter](#)).

### 35.4.4 eTPU channel output disable signals

Each of these 4 input signals are used to force the outputs of a group of 8 channels to an inactive level. When an ODIS input is active, all the channels in its group of 8 that have their fields ODIS=1 in [eTPU Channel n Configuration Register for Engine 1 \(eTPU\\_CCCR\\_CnCR\\_ENG1\)](#) and [eTPU Channel n Configuration Register for Engine 2](#)

2. Sampled on the T4 microcycle phase, see [Microcycle and I/O timing](#).

([eTPU\\_CCCR\\_CnCR\\_ENG2](#)) have their outputs forced to the opposite of the value specified in bit OPOL of the same register. Therefore, channels can be individually selected to be affected by the output disable signals, as well as their disabling forced polarity (see 'Pin State Input/Output Logic' figure in [Pin Control Registers](#)).

See the chip-specific eTPU information for details on the output channel disable group definition.

In a Dual-engine eTPU there are 8 output disable signals for the 64 channels.

## 35.5 Memory map/register definition

The guideline for the description of all bits and fields throughout the “Memory Map/ Register Definition” section is to provide only a brief explanation (without examples or method of use) of the features, since it will be used mainly as a reference for the reader that is studying the “Functional Description” section, where those features are explained in detail.

## 35.6 Memory map

The eTPU System simplified memory map is shown in [Table 35-4](#). Each of the register areas shown may have their own reserved address areas.

[Table 35-5](#) shows a detailed memory map. Offsets are relative to the eTPU Base address, which is MCU-dependent.

See the chip-specific eTPU information about accessing eTPU registers in chip-level Stop mode.

SCM unused area is decoded and returns a fixed opcode, determined by [eTPU SCM Off-range Data Register \(eTPU\\_SCR\\_SCMOFFDATAR\)](#).

**Table 35-4. High-level memory map**

| Offset      | Use                            |
|-------------|--------------------------------|
| 0x00 – 0x1F | System Configuration Registers |
| 0x20 – 0x2F | eTPU 1 Time Base Registers     |
| 0x30 – 0x3F | Reserved <sup>1</sup>          |
| 0x40 – 0x4F | eTPU 2 Time Base Registers     |
| 0x50 – 0x5F | Reserved <sup>1</sup>          |
| 0x60 – 0x6F | eTPU 1 Extra Engine Registers  |
| 0x70 – 0x7F | eTPU 2 Extra Engine Registers  |

*Table continues on the next page...*

**Table 35-4. High-level memory map (continued)**

| Offset                         | Use                                     |
|--------------------------------|-----------------------------------------|
| 0x80 – 0xFF                    | Latency Monitor Registers               |
| 0x100 – 0x13F                  | Memory Error Support Registers          |
| 0x140 – 0x1FF                  | Reserved <sup>1</sup>                   |
| 0x200 – 0x2FF                  | eTPU 1/2 Global Channel Registers       |
| 0x300 – 0x3FF                  | Reserved <sup>1</sup>                   |
| 0x400 – 0x7FF                  | eTPU 1 Channel Registers                |
| 0x800 – 0xBFF                  | eTPU 2 Channel Registers                |
| 0xC00 – 0xFFF                  | Shared Subsystems Integration Registers |
| 0x1000 – 0x3FFF                | Reserved <sup>1</sup>                   |
| 0x4000 – 0x7FFF                | SSS - Shared Subsystems area            |
| 0x8000 – 0xBFFF <sup>2</sup>   | SDM - Shared Data Memory                |
| 0xC000 – 0xFFFF <sup>2</sup>   | SDM PSE mirror <sup>3</sup>             |
| 0x10000 – 0x1FFFF <sup>2</sup> | SCM - Shared Code Memory <sup>4</sup>   |

1. Reserved addresses must not be used. Access to these memory positions complete with 0-wait-states, but the result is unspecified.
2. Actual sizes of SCM and SDM are MCU-dependent (see the chip-specific eTPU information).
3. Parameter Sign Extension access area, see [Parameter Access](#).
4. SCM access is available only when bit VIS=1 on register eTPU\_SCR\_MCR, under certain conditions (see [MCR](#)).

**Table 35-5. Detailed memory map**

| Offset                         | Use                                                                  |
|--------------------------------|----------------------------------------------------------------------|
| System Configuration Registers |                                                                      |
| 0x00                           | eTPU_SCR_MCR - eTPU Module Configuration Register                    |
| 0x04                           | eTPU_SCR_CDCR - eTPU Coherent Dual-Parameter Controller Register     |
| 0x08                           | Reserved                                                             |
| 0x0C                           | eTPU_SCR_MISCCMPR - eTPU MISC Compare Register                       |
| 0x10                           | eTPU_SCR_SCMOFFDATAR - eTPU SCM Off-range Data Register <sup>1</sup> |
| 0x14                           | eTPU_SCR_ECR_ENG1 - eTPU 1 Engine Configuration Register             |
| 0x18                           | eTPU_SCR_ECR_ENG2 - eTPU 2 Engine Configuration Register             |
| 0x1C                           | Reserved                                                             |
| Time Base Registers            |                                                                      |
| 0x20                           | eTPU_TBR_TBCR_ENG1 - eTPU 1 Time Base Configuration Register         |
| 0x24                           | eTPU_TBR_TB1R_ENG1 - eTPU 1 Time Base 1                              |
| 0x28                           | eTPU_TBR_TB2R_ENG1 - eTPU 1 Time Base 2                              |
| 0x2C                           | eTPU_TBR_STAC_ENG1 - eTPU 1 STAC Configuration Register              |
| 0x30                           | Reserved                                                             |
| 0x34                           | Reserved                                                             |
| 0x38                           | Reserved                                                             |
| 0x3C                           | Reserved                                                             |
| 0x40                           | eTPU_TBR_TBCR_ENG2 - eTPU 2 Time Base Configuration Register         |

Table continues on the next page...



**Table 35-5. Detailed memory map (continued)**

| Offset                                      | Use                                                                   |
|---------------------------------------------|-----------------------------------------------------------------------|
| 0x44                                        | eTPU_TBR_TB1R_ENG2 - eTPU 2 Time Base 1                               |
| 0x48                                        | eTPU_TBR_TB2R_ENG2 - eTPU 2 Time Base 2                               |
| 0x4C                                        | eTPU_TBR_STAC_ENG2 - eTPU 2 STAC Configuration Register               |
| 0x50                                        | Reserved                                                              |
| 0x54                                        | Reserved                                                              |
| 0x58                                        | Reserved                                                              |
| 0x5C                                        | Reserved                                                              |
| Engine Related Registers                    |                                                                       |
| 0x60                                        | eTPU_ER_WDTR_ENG1 - eTPU 1 Watchdog Timer Register                    |
| 0x64                                        | Reserved                                                              |
| 0x68                                        | eTPU_ER_IDLER_ENG1 - eTPU 1 Idle Counter Register                     |
| 0x6C                                        | Reserved                                                              |
| 0x70                                        | eTPU_ER_WDTR_ENG2 - eTPU 2 Watchdog Timer Register                    |
| 0x74                                        | Reserved                                                              |
| 0x78                                        | eTPU_ER_IDLER_ENG2 - eTPU 2 Idle Counter Register                     |
| 0x7C                                        | Reserved                                                              |
| Latency Monitor Registers                   |                                                                       |
| 0x80                                        | eTPU_LMR_LM0CR_ENG1 - eTPU 1 Latency Monitor 0 Configuration Register |
| 0x84                                        | eTPU_LMR_LM0MR_ENG1 - eTPU 1 Latency Monitor 0 Measure Register       |
| 0x88                                        | eTPU_LMR_LM1CR_ENG1 - eTPU 1 Latency Monitor 1 Configuration Register |
| 0x8C                                        | eTPU_LMR_LM1MR_ENG1 - eTPU 1 Latency Monitor 1 Measure Register       |
| 0xC0                                        | eTPU_LMR_LM0CR_ENG2 - eTPU 2 Latency Monitor 0 Configuration Register |
| 0xC4                                        | eTPU_LMR_LM0MR_ENG2 - eTPU 2 Latency Monitor 0 Measure Register       |
| 0xC8                                        | eTPU_LMR_LM1CR_ENG2 - eTPU 2 Latency Monitor 1 Configuration Register |
| 0xCC                                        | eTPU_LMR_LM1MR_ENG2 - eTPU 2 Latency Monitor 1 Measure Register       |
| Memory Error Support Registers <sup>2</sup> |                                                                       |
| 0x100                                       | eTPU_MESR_MECSR - eTPU Memory Error Control Register                  |
| 0x104                                       | eTPU_MESR_DEIAR - eTPU Data Error Injection Address Register          |
| 0x108                                       | eTPU_MESR_DEIDPR - eTPU Data Error Injection Data Pattern Register    |
| 0x10C                                       | eTPU_MESR_DEIPPR - eTPU Data Error Injection Parity Pattern Register  |
| 0x110                                       | eTPU_MESR_DERAR - eTPU Data Error Report Address Register             |
| 0x114                                       | eTPU_MESR_DERDR - eTPU Data Error Report Data Register                |
| 0x118                                       | eTPU_MESR_DERSR - eTPU Data Error Report Syndrome Register            |
| 0x11C                                       | Reserved                                                              |
| 0x120                                       | eTPU_MESR_MESR - eTPU Memory Error Status Register                    |
| 0x124                                       | eTPU_MESR_CEIAR - eTPU Code Error Injection Address Register          |
| 0x128                                       | eTPU_MESR_CeidPR - eTPU Code Error Injection Data Pattern Register    |
| 0x12C                                       | eTPU_MESR_CeiPPR - eTPU Code Error Injection Parity Pattern Register  |
| 0x130                                       | eTPU_MESR_CERAR - eTPU Code Error Report Address Register             |

Table continues on the next page...



**Table 35-5. Detailed memory map (continued)**

| Offset                   | Use                                                                                   |
|--------------------------|---------------------------------------------------------------------------------------|
| 0x134                    | eTPU_MESR_CERDR - eTPU Code Error Report Data Register                                |
| 0x138                    | eTPU_MESR_CERSR - eTPU Code Error Report Syndrome Register                            |
| 0x13C                    | eTPU_MESR_CEFR - eTPU Code Error Fix Register                                         |
| 0x140 - 0x1FF            | Reserved                                                                              |
| Global Channel Registers |                                                                                       |
| 0x200                    | eTPU_GCR_CISR_ENG1 - eTPU 1 Channel Interrupt Status Register                         |
| 0x204                    | eTPU_GCR_CISR_ENG2 - eTPU 2 Channel Interrupt Status Register                         |
| 0x208                    | Reserved                                                                              |
| 0x20C                    | Reserved                                                                              |
| 0x210                    | eTPU_GCR_CDTRSR_ENG1 - eTPU 1 Channel Data Transfer Request Status Register           |
| 0x214                    | eTPU_GCR_CDTRSR_ENG2 - eTPU 2 Channel Data Transfer Request Status Register           |
| 0x218                    | Reserved                                                                              |
| 0x21C                    | Reserved                                                                              |
| 0x220                    | eTPU_GCR_CIOSR_ENG1 - eTPU 1 Channel Interrupt Overflow Status Register               |
| 0x224                    | eTPU_GCR_CIOSR_ENG2 - eTPU 2 Channel Interrupt Overflow Status Register               |
| 0x228                    | Reserved                                                                              |
| 0x22C                    | Reserved                                                                              |
| 0x230                    | eTPU_GCR_CDTROSR_ENG1 - eTPU 1 Channel Data Transfer Request Overflow Status Register |
| 0x234                    | eTPU_GCR_CDTROSR_ENG2 - eTPU 2 Channel Data Transfer Request Overflow Status Register |
| 0x238                    | Reserved                                                                              |
| 0x23C                    | Reserved                                                                              |
| 0x240                    | eTPU_GCR_CIER_ENG1 - eTPU 1 Channel Interrupt Enable Register                         |
| 0x244                    | eTPU_GCR_CIER_ENG2 - eTPU 2 Channel Interrupt Enable Register                         |
| 0x248                    | Reserved                                                                              |
| 0x24C                    | Reserved                                                                              |
| 0x250                    | eTPU_GCR_CDTREER_ENG1 - eTPU 1 Channel Data Transfer Request Enable Register          |
| 0x254                    | eTPU_GCR_CDTREER_ENG2 - eTPU 2 Channel Data Transfer Request Enable Register          |
| 0x258                    | Reserved                                                                              |
| 0x25C                    | Reserved                                                                              |
| 0x260                    | eTPU_GCR_WDSR_ENG1 - eTPU 1 Watchdog Status Register                                  |
| 0x264                    | eTPU_GCR_WDSR_ENG2 - eTPU 2 Watchdog Status Register                                  |
| 0x268-0x27F              | Reserved                                                                              |
| 0x280                    | eTPU_GCR_CPSSR_ENG1 - eTPU 1 Channel Pending Service Status Register                  |
| 0x284                    | eTPU_GCR_CPSSR_ENG2 - eTPU 2 Channel Pending Service Status Register                  |
| 0x288                    | Reserved                                                                              |
| 0x28C                    | Reserved                                                                              |
| 0x290                    | eTPU_GCR_CSSR_ENG1 - eTPU 1 Channel Service Status Register                           |
| 0x294                    | eTPU_GCR_CSSR_ENG2 - eTPU 2 Channel Service Status Register                           |
| 0x298                    | Reserved                                                                              |

*Table continues on the next page...*

**Table 35-5. Detailed memory map (continued)**

| Offset                                      | Use                                                                      |
|---------------------------------------------|--------------------------------------------------------------------------|
| 0x29C                                       | Reserved                                                                 |
| 0x300 - 0x3FF                               | Reserved                                                                 |
| Channel Configuration and Control Registers |                                                                          |
| 0x400                                       | eTPU_CCCR_C0CR_ENG1 - eTPU 1 Channel 0 Configuration Register            |
| 0x404                                       | eTPU_CCCR_C0SCR_ENG1 - eTPU 1 Channel 0 Status and Control Register      |
| 0x408                                       | eTPU_CCCR_C0HSRR_ENG1 - eTPU 1 Channel 0 Host Service Request Register   |
| 0x40C                                       | Reserved                                                                 |
| 0x410                                       | eTPU_CCCR_C1CR_ENG1 - eTPU 1 Channel 1 Configuration Register            |
| 0x414                                       | eTPU_CCCR_C1SCR_ENG1 - eTPU 1 Channel 1 Status and Control Register      |
| 0x418                                       | eTPU_CCCR_C1HSRR_ENG1 - eTPU 1 Channel 1 Host Service Request Register   |
| 0x41C                                       | Reserved                                                                 |
| -----                                       | -----                                                                    |
| 0x5F0                                       | eTPU_CCCR_C31CR_ENG1 - eTPU 1 Channel 31 Configuration Register          |
| 0x5F4                                       | eTPU_CCCR_C31SCR_ENG1 - eTPU 1 Channel 31 Status and Control Register    |
| 0x5F8                                       | eTPU_CCCR_C31HSRR_ENG1 - eTPU 1 Channel 31 Host Service Request Register |
| 0x5FC - 0x7FF                               | Reserved                                                                 |
| 0x800                                       | eTPU_CCCR_C0CR_ENG2 - eTPU 2 Channel 0 Configuration Register            |
| 0x804                                       | eTPU_CCCR_C0SCR_ENG2 - eTPU 2 Channel 0 Status and Control Register      |
| 0x808                                       | eTPU_CCCR_C0HSRR_ENG2 - eTPU 2 Channel 0 Host Service Request Register   |
| 0x80C                                       | Reserved                                                                 |
| 0x810                                       | eTPU_CCCR_C1CR_ENG2 - eTPU 2 Channel 1 Configuration Register            |
| 0x814                                       | eTPU_CCCR_C1SCR_ENG2 - eTPU 2 Channel 1 Status and Control Register      |
| 0x818                                       | eTPU_CCCR_C1HSRR_ENG2 - eTPU 2 Channel 1 Host Service Request Register   |
| 0x81C                                       | Reserved                                                                 |
| -----                                       | -----                                                                    |
| 0x9F0                                       | eTPU_CCCR_C31CR_ENG2 - eTPU 2 Channel 31 Configuration Register          |
| 0x9F4                                       | eTPU_CCCR_C31SCR_ENG2 - eTPU 2 Channel 31 Status and Control Register    |
| 0x9F8                                       | eTPU_CCCR_C31HSRR_ENG2 - eTPU 2 Channel 31 Host Service Request Register |
| 0x9FC - 0xBFF                               | Reserved                                                                 |
| Shared Subsystem Support Registers          |                                                                          |
| 0xC00                                       | eTPU_SSIR_SSS0IR - eTPU SSS 0 Integration Register                       |
| 0xC04                                       | eTPU_SSIR_SSS1IR - eTPU SSS 1 Integration Register                       |
| 0xC08                                       | eTPU_SSIR_SSS2IR - eTPU SSS 2 Integration Register                       |
| 0xC0A                                       | eTPU_SSIR_SSS3IR - eTPU SSS 3 Integration Register                       |
| 0xC0C                                       | eTPU_SSIR_SSS4IR - eTPU SSS 4 Integration Register                       |
| 0xC10                                       | eTPU_SSIR_SSS5IR - eTPU SSS 5 Integration Register                       |
| 0xC14                                       | eTPU_SSIR_SSS6IR - eTPU SSS 6 Integration Register                       |
| 0xC18                                       | eTPU_SSIR_SSS7IR - eTPU SSS 7 Integration Register                       |
| 0xC1A                                       | eTPU_SSIR_SSS8IR - eTPU SSS 8 Integration Register                       |

Table continues on the next page...

**Table 35-5. Detailed memory map (continued)**

| Offset                       | Use                                                |
|------------------------------|----------------------------------------------------|
| 0xC1C                        | eTPU_SSIR_SSS9IR - eTPU SSS 9 Integration Register |
| 0x1000-0x3FFF                | Reserved                                           |
| 0x4000 - 0x7FFF              | Shared Subsystems area - SSS                       |
| 0x8000 - 0xBFFF <sup>3</sup> | Shared Data Memory - SDM                           |
| 0xC000 - 0xFFFF <sup>3</sup> | Shared Data Memory - SDM - PSE mirror <sup>4</sup> |
| 0x10000-1FFFF <sup>5</sup>   | Shared Code Memory - SCM <sup>6</sup>              |

1. This register is not implemented in some MCUs; see [SCMOFFDATAR](#).
2. These registers are not implemented in some MCUs; see [Memory error support](#).
3. The actual SDM size is MCU-dependent (see the chip-specific eTPU information). SDM is allocated from the starting value of the given range of offsets.
4. Parameter Sign Extension access area, see [Parameter Access](#).
5. The actual SCM size is MCU-dependent (see the chip-specific eTPU information). SCM is allocated from the starting value of the given range of offsets. When the size is not the maximum, the unused SCM address range returns the value of [SCMOFFDATAR](#).
6. SCM access is available only when eTPU\_SCR\_MCR[VIS]=1, under certain conditions (see [MCR](#)).

## 35.7 System Configuration memory map and registers

### eTPU\_SCR memory map

| Address offset (hex) | Register name                                                    | Width (in bits) | Access | Reset value                 | Section/ page               |
|----------------------|------------------------------------------------------------------|-----------------|--------|-----------------------------|-----------------------------|
| 0                    | eTPU Module Configuration Register (eTPU_SCR_MCR)                | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.7.1/1304</a> |
| 4                    | eTPU Coherent Dual-Parameter Controller Register (eTPU_SCR_CDCR) | 32              | R/W    | 0000_0000h                  | <a href="#">35.7.2/1308</a> |
| C                    | eTPU MISC Compare Register (eTPU_SCR_MISCCMPR)                   | 32              | R/W    | 0000_0000h                  | <a href="#">35.7.3/1310</a> |
| 10                   | eTPU SCM Off-range Data Register (eTPU_SCR_SCMOFFDATAR)          | 32              | R/W    | F377_5FFBh                  | <a href="#">35.7.4/1310</a> |
| 14                   | eTPU Engine 1 Configuration Register (eTPU_SCR_ECR_ENG1)         | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.7.5/1311</a> |
| 18                   | eTPU Engine 2 Configuration Register (eTPU_SCR_ECR_ENG2)         | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.7.6/1315</a> |

### 35.7.1 eTPU Module Configuration Register (eTPU\_SCR\_MCR)

This register is global to both eTPU engines, and resides in the shared BIU. It gathers global configuration and status in the eTPU system, including global exception. It is also used for configuring the SCM (shared code memory) operation and test.

Address: 0h base + 0h offset = 0h

| Bit   | 0   | 1      | 2     | 3     | 4    | 5    | 6    | 7    | 8      | 9 | 10 | 11      | 12 | 13 | 14 | 15 |
|-------|-----|--------|-------|-------|------|------|------|------|--------|---|----|---------|----|----|----|----|
| R     | 0   | SDMERR | WDTO1 | WDTO2 | MGE1 | MGE2 | ILF1 | ILF2 | SSAERR | 0 | 0  | SCMSIZE |    |    |    |    |
| W     | GEC |        |       |       |      |      |      |      |        |   |    |         |    |    |    |    |
| Reset | 0   | 0      | 0     | 0     | 0    | 0    | 0    | 0    | 0      | 0 | 0  | *       | *  | *  | *  | *  |

| Bit   | 16   | 17   | 18 | 19 | 20      | 21      | 22       | 23     | 24 | 25  | 26 | 27 | 28 | 29 | 30   | 31 |
|-------|------|------|----|----|---------|---------|----------|--------|----|-----|----|----|----|----|------|----|
| R     | LTO1 | LTO2 | 0  |    | SCMMISC | SCMMISF | SCMMISEN | SCMERR | 0  | VIS | 0  |    |    |    | GTBE |    |
| W     |      |      |    |    |         |         |          |        |    |     |    |    |    |    |      |    |
| Reset | 0    | 0    | 0  | 0  | 0       | 0       | 0        | 0      | 0  | 0   | 0  | 0  | 0  | 0  | 0    | 0  |

\* Notes:

- SCMSIZE field: See the chip-specific eTPU information for this field's reset value.

**eTPU\_SCR\_MCR field descriptions**

| Field       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>GEC    | <p>Global Exception Clear.</p> <p>This write-only bit negates global exception request and clears global exception status bits SDMERR, SSAERR, SCMERR, WDTO1/2, MGE1/2, ILF1/2, LTO1/2, and SCMMISF.</p> <p>GEC works the same way with either one or both Engines in Module Disable Mode.</p> <p>0 Keep Global Exception request and status bits SDMERR, SSAERR, SCMERR, WDTO1/2, ILF1/2, MGE1/2, LTO1/2, and SCMMISF as is.</p> <p>1 Negate Global Exception, clear status bits SDMERR, SSAERR, SCMERR, WDTO1/2, ILF1/2, MGE1/2, LTO1/2, and SCMMISF.</p> |
| 1<br>SDMERR | <p>SDM Read Error</p> <p>This flag indicates that an SDM non-correctable read error occurred on a microengine or host (including CDC) read, generating a Global Exception. This bit is cleared by writing 1 to GEC.</p> <p>0 No Global Exception pending because of SDM read error.</p> <p>1 Global Exception requested by SDM read error is pending.</p>                                                                                                                                                                                                   |
| 2<br>WDTO1  | <p>Watchdog 1 Timeout.</p> <p>Flag WDTO1 indicates that a Watchdog Timeout occurred in engine 1, generating a Global Exception. This bit is cleared by writing 1 to GEC.</p> <p>0 No Global Exception pending because of Watchdog timeout.</p> <p>1 Global Exception requested by Watchdog timeout.</p>                                                                                                                                                                                                                                                     |
| 3<br>WDTO2  | <p>Watchdog 2 Timeout.</p> <p>Flag WDTO2 indicates that a Watchdog Timeout occurred in engine 2, generating a Global Exception. This bit is cleared by writing 1 to GEC.</p> <p>0 No Global Exception pending because of Watchdog timeout.</p> <p>1 Global Exception requested by Watchdog timeout.</p>                                                                                                                                                                                                                                                     |
| 4<br>MGE1   | <p>Microcode Global Exception Engine 1.</p> <p>This bit indicates that a global exception was asserted by microcode executed on engine 1. The determination of the reason why the global exception was asserted is application dependent: it can be coded in an SDM status parameter, for instance. This bit is cleared by writing 1 to GEC.</p> <p>0 No microcode-requested global exception pending.</p> <p>1 Global exception requested by microcode is pending.</p>                                                                                     |
| 5<br>MGE2   | <p>Microcode Global Exception Engine 2.</p> <p>This bit indicates that a global exception was asserted by microcode executed on engine 2. The determination of the reason why the global exception was asserted is application dependent: it can be coded in an SDM status parameter, for instance. This bit is cleared by writing 1 to GEC.</p> <p>0 No microcode-requested global exception pending.</p> <p>1 Global exception requested by microcode is pending.</p>                                                                                     |
| 6<br>ILF1   | <p>Illegal Instruction Flag for Engine 1.</p> <p>The ILFA1 flag is set by the microengine to indicate that an illegal instruction was decoded in Engine 1. This bit is cleared by host writing 1 to GEC. See <a href="#">Illegal instructions</a> for more details.</p>                                                                                                                                                                                                                                                                                     |

*Table continues on the next page...*

## eTPU\_SCR\_MCR field descriptions (continued)

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | 0 Illegal Instruction not detected.<br>1 Illegal Instruction detected by Engine 1.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 7<br>ILF2         | Illegal Instruction Flag for Engine 2.<br>The ILFA2 flag is set by the microengine to indicate that an illegal instruction was decoded in Engine 2. This bit is cleared by host writing 1 to GEC. See <a href="#">Illegal instructions</a> for more details.<br>0 Illegal Instruction not detected.<br>1 Illegal Instruction detected by Engine 2.                                                                                                                                                                                                                   |
| 8<br>SSAERR       | Shared Subsystem Access Error.<br>This flag indicates that a bus error occurred in an microengine access to the Shared Subsystem Area (see <a href="#">Shared Subsystems support</a> ). This bit is cleared by writing 1 to GEC.<br><b>NOTE:</b> Bus errors on host accesses to the Shared Subsystem Area do not set this bit neither asserts the Global Exception, but the bus error is propagated to the host CPU<br>0 No bus error returned on a microengine Shared Subsystem Area access.<br>1 Bus error returned on a microengine Shared Subsystem Area access. |
| 9<br>Reserved     | Reserved.<br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 10<br>Reserved    | Reserved.<br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 11–15<br>SCMSIZE  | SCM Size<br>This read-only field holds the number of 2 KB SCM Blocks minus 1. For more information, see the MCU Reference Manual.<br>See the chip-specific eTPU information for this field's reset value.                                                                                                                                                                                                                                                                                                                                                            |
| 16<br>LTO1        | Latency Timeout for Engine 1.<br>The LTO1 flag indicates the latency measured by the Latency Monitor of Engine 1 surpassed the specified maximum. See <a href="#">Latency Monitor</a> for details. This bit is cleared by writing 1 to GEC.<br>0 No latency timeout detected by Engine 1's Latency Monitor.<br>1 Latency timeout detected by Engine 1's Latency Monitors.                                                                                                                                                                                            |
| 17<br>LTO2        | Latency Timeout for Engine 2.<br>The LTO2 flag indicates the latency measured by the Latency Monitor of Engine 2 surpassed the specified maximum. See <a href="#">Latency Monitor</a> for details. This bit is cleared by writing 1 to GEC.<br>0 No latency timeout detected by Engine 2's Latency Monitor.<br>1 Latency timeout detected by Engine 2's Latency Monitors.                                                                                                                                                                                            |
| 18–19<br>Reserved | Reserved.<br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

Table continues on the next page...

**eTPU\_SCR\_MCR field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 20<br>SCMMISC  | <p>SCM MISC Complete flag.</p> <p>Flag SCMMISC indicates that MISC has completed the evaluation of the SCM signature since reset or since the last time it was cleared. SCMMISC is cleared by writing 1 to it, and is not cleared when MISC is disabled (SCMMISEN=0). SCMMISC asserts at the end of the SCM memory scan, either if the signature matches or not.</p> <p>0 MISC has not yet completed an SCM signature calculation and compare since the last time SCMMISC was cleared.</p> <p>1 MISC completed at least one SCM signature calculation and compare since the last time SCMMISC was cleared.</p>                                                                                                                                                                                                                                                    |
| 21<br>SCMMISF  | <p>SCM MISC Flag.</p> <p>The SCMMISF bit is set by the SCM MISC (Multiple Input Signature Calculator) logic to indicate that the calculated signature does not match the expected value, at the end of a MISC iteration. See <a href="#">SCM Test - Multiple Input Signature Calculator</a> for more details.</p> <p>This bit is cleared when Global Exception is cleared by writing 1 to GEC.</p> <p>0 Signature mismatch not detected.</p> <p>1 MISC has read entire SCM array and the expected signature in <a href="#">eTPU MISC Compare Register (eTPU_SCR_MISCCMPR)</a> does not match the value calculated.</p>                                                                                                                                                                                                                                            |
| 22<br>SCMMISEN | <p>SCM MISC Enable.</p> <p>The SCMMISEN bit is used for enabling/disabling the operation of the MISC logic. SCMMISEN is readable and writable at any time. The MISC logic will only operate when this bit is set to 1. When the bit is reset, the MISC address counter is set to the initial SCM address. When enabled, the MISC will continuously cycle through the SCM addresses, reading each and calculating a CRC. In order to save power, the MISC can be disabled by clearing the SCMMISEN bit. See <a href="#">SCM Test - Multiple Input Signature Calculator</a> for more details.</p> <p>SCMMISEN resets automatically when MISC logic detects an error, i.e., when SCMMISF transitions from 0 to 1, disabling the MISC operation.</p> <p>0 MISC operation disabled. The MISC logic is reset to its initial state.</p> <p>1 MISC operation enabled.</p> |
| 23<br>SCMERR   | <p>SCM Read Error.</p> <p>This flag indicates that an SCM non-correctable read error occurred on a host or microengine (including MISC) read, generating a Global Exception. This bit is cleared by writing 1 to GEC.</p> <p>0 No Global Exception pending because of SCM read error.</p> <p>1 Global Exception requested by SCM read error is pending.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 24<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 25<br>VIS      | <p>SCM Visibility Bit.</p> <p>VIS bit turns SCM visible to the IP-Bus and resets MISC state (but SCMMISEN keeps its value).</p> <p>This bit is write protected when any of the Engines is not halted or stopped.</p> <p>When VIS=1, the MDIS bits in the eTPU Memory Error Support registers are write protected.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

*Table continues on the next page...*

**eTPU\_SCR\_MCR field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | 0 SCM is not visible to the slave bus. Accessing SCM address space issues a bus error, writes are protected and reads are meaningless.<br>1 SCM is visible to the slave bus. MISC state is reset.                                                                                                                                                                                                                                                                                                                                                                                                    |
| 26–30<br>Reserved | Reserved.<br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 31<br>GTBE        | Global Time Base Enable.<br>GTBE enables time bases in both Engines, allowing them to be started synchronously.<br><b>NOTE:</b> Global Time Base Enable action may also depend on other blocks, as explained in <a href="#">GTBE - Global Time Base Enable</a> .<br><b>NOTE:</b> When GTBE is turned off with Angle Mode enabled, the EAC must be reinitialized before GTBE is turned on again. The EAC reinitialization procedure is described in <a href="#">Restarting angle logic</a> .<br>0 Time bases in both Engines are disabled to run.<br>1 Time bases in both Engines are enabled to run. |

## 35.7.2 eTPU Coherent Dual-Parameter Controller Register (eTPU\_SCR\_CDCR)

This register configures and controls dual-parameter coherent transfers. For more information, see [Coherent Dual-parameter Controller \(CDC\)](#).

Address: 0h base + 4h offset = 4h

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
| R     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |



## eTPU\_SCR\_CDCR field descriptions

| Field          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>STS       | <p>Start Bit.</p> <p>This bit is set by the host in order to start the data transfer between the parameter buffer pointed by PBBASE and the target addresses selected by the concatenation of fields CTBASE and PARM0/1. The host receives wait-states until the data transfer is complete, when this bit is reset by the coherency logic (see <a href="#">Coherent Dual-parameter Controller (CDC)</a>). Therefore, host always reads STS as 0.</p> <p>0 Write does not start a coherent transfer.<br/>1 Write starts a coherent transfer.</p> |
| 1–5<br>CTBASE  | <p>Channel Transfer Base.</p> <p>This field concatenates with fields PARM0 and PARM1 to determine the absolute word offset (from the SDM base) of the parameters to be transferred:</p> <p>Parameter 0 word address = {CTBASE, PARM0} + SDM base word address.<br/>Parameter 1 word address = {CTBASE, PARM1} + SDM base word address.</p>                                                                                                                                                                                                      |
| 6–15<br>PBBASE | <p>Parameter Buffer Base Address.</p> <p>This field points to the base address of the parameter buffer location, with granularity of 2 parameters (8 bytes). The host (byte) address of the first parameter in the buffer is PBBASE * 8 + SDM Base Byte Address. The microengine absolute (word) address of the first parameter in the buffer is PBBASE * 2.</p>                                                                                                                                                                                |
| 16<br>PWIDTH   | <p>Parameter Width Selection.</p> <p>This bit selects the width of the parameters to be transferred between the PB and the target address.</p> <p>0 Transfer 24-bit parameters. The upper byte remains unchanged in the destination address.<br/>1 Transfer 32-bit parameters. All 32 bits of the parameters are written in the destination address.</p>                                                                                                                                                                                        |
| 17–23<br>PARM0 | <p>Channel Parameter number 0.</p> <p>The PARM0 and PARM1 fields in concatenation with CTBASE determine the word address offset (from the SDM base) of the parameters that are destination or source (defined by WR) of the coherent transfer. The word SDM address offset of the parameters are {CTBASE, PARM0/1}.</p> <p><b>NOTE:</b> PARM0 and PARM1 allow non-contiguous parameters to be transferred coherently. The parameter pointed by {CTBASE, PARM0} is the first transferred.</p>                                                    |
| 24<br>WR       | <p>Read/Write selection.</p> <p>This bit selects the direction of the coherent data transfer.</p> <p>0 Read operation. Data transfer is from the selected SDM address to the PB.<br/>1 Write operation. Data transfer is from the PB to the selected SDM address.</p>                                                                                                                                                                                                                                                                           |
| 25–31<br>PARM1 | <p>Channel Parameter number 1.</p> <p>The PARM0 and PARM1 fields in concatenation with CTBASE determine the word address offset (from the SDM base) of the parameters that are destination or source (defined by WR) of the coherent transfer. The word SDM address offset of the parameters are {CTBASE, PARM0/1}.</p> <p><b>NOTE:</b> PARM0 and PARM1 allow non-contiguous parameters to be transferred coherently. The parameter pointed by {CTBASE, PARM0} is the first transferred.</p>                                                    |

### 35.7.3 eTPU MISC Compare Register (eTPU\_SCR\_MISCCMPR)

This register holds the 32-bit signature expected from the whole SCM array. This register must be written by the host with the 32-bit word to be compared against the calculated signature at the end of the MISC cycle. This register is global to both eTPU Engines. For more detail, see [SCM Test - Multiple Input Signature Calculator](#).

Address: 0h base + Ch offset = Ch

|       |             |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0           | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | ETPUMISCCMP |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |             |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0           | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

#### eTPU\_SCR\_MISCCMPR field descriptions

| Field               | Description                                                                                                                            |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>ETPUMISCCMP | Expected Multiple Input Signature Register value.<br>For details, see <a href="#">SCM Test - Multiple Input Signature Calculator</a> . |

### 35.7.4 eTPU SCM Off-range Data Register (eTPU\_SCR\_SCMOFFDATAR)

This register holds the 32-bit value returned when SCM array is accessed at non-implemented addresses, either by the Host or by the microengine. This register can be written by the host with the 32-bit instruction to be executed by the microengine to recover from runaway code. This register is global to both eTPU Engines. For more detail see [SCM Off-range data](#).

Address: 0h base + 10h offset = 10h

|       |                 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-----------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0               | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | ETPUSCMOFFDATAR |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |                 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 1               | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1  | 1  | 0  | 1  | 1  | 1  | 0  | 1  | 0  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 0  | 1  | 1  |

#### eTPU\_SCR\_SCMOFFDATAR field descriptions

| Field                   | Description                                                                |
|-------------------------|----------------------------------------------------------------------------|
| 0–31<br>ETPUSCMOFFDATAR | SCM Off-range read data value.<br>See <a href="#">SCM Off-range data</a> . |

### 35.7.5 eTPU Engine 1 Configuration Register (eTPU\_SCR\_ECR\_ENG1)

This register holds configuration and status fields for engine 1.

#### NOTE

See Register Reset Values in the chip-specific eTPU information.

Address: 0h base + 14h offset = 14h

| Bit   | 0    | 1    | 2  | 3    | 4  | 5  | 6  | 7  | 8      | 9  | 10 | 11 | 12  | 13   | 14    | 15 |
|-------|------|------|----|------|----|----|----|----|--------|----|----|----|-----|------|-------|----|
| R     | FEND | MDIS | 0  | STF  | 0  |    |    |    | HLTF   | 0  |    |    |     | FCSS | FPSCK |    |
| W     |      |      |    |      |    |    |    |    |        |    |    |    |     |      |       |    |
| Reset | 0    | *    | 0  | 0    | 0  | 0  | 0  | 0  | *      | 0  | 0  | 0  | 0   | 0    | 0     | 0  |
| Bit   | 16   | 17   | 18 | 19   | 20 | 21 | 22 | 23 | 24     | 25 | 26 | 27 | 28  | 29   | 30    | 31 |
| R     | CDFC |      | 0  | ERBA |    |    |    |    |        |    |    | 0  | ETB |      |       |    |
| W     |      |      |    |      |    |    |    |    | SPPDIS |    |    |    |     |      |       |    |
| Reset | 0    | 0    | 0  | 0    | 0  | 0  | 0  | 0  | 0      | 0  | 0  | 0  | 0   | 0    | 0     | 0  |

\* Notes:

- HLTF field: HLTF field reset value is 1 if, after a reset, the eTPU Nexus debug support is on and a breakpoint request is asserted. Otherwise the reset value is 0.
- MDIS field: Reset value is 0.

#### eTPU\_SCR\_ECR\_ENG1 field descriptions

| Field     | Description                                                                                                                                                            |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>FEND | Force End.<br><br>FEND assertion terminates any current running thread as if an END instruction have been executed (see <a href="#">Ending Current Thread - END</a> ). |

Table continues on the next page...

**eTPU\_SCR\_ECR\_ENG1 field descriptions (continued)**

| Field         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               | <p>This bit is self-negating when the thread ends.</p> <p><b>NOTE:</b> Only on rare occasions (e.g., during a long stall, see <a href="#">Microengine Stall</a>) FEND can be read as 1, because it negates as soon as the end begins execution.</p> <p>Writing FEND = 1 is ignored and FEND stays 0 when the microengine is in TST, halted, stopped, or idle (no thread executing).</p> <p>0 Normal operation.</p> <p>1 Ends any ongoing thread.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 1<br>MDIS     | <p>Module Disable Bit.</p> <p>When MDIS is set, the Engine shuts down its internal clocks, going into Module Disable Mode. TCR1 and TCR2 cease to increment, and input sampling stops. The Engine asserts the stop flag (STF) bit to indicate that it has stopped. However, the BIU continues to run, and the Host can access all registers except for the channel registers (see channel registers starting with <a href="#">eTPU Channel n Configuration Register for Engine 1 (eTPU_CCCR_CnCR_ENG1)</a>)</p> <p><b>NOTE:</b> The Timebase registers can still be read with MDIS=1, but writes are ineffective and a Bus Error is issued. Global Channel Registers and SDM can be accessed normally.</p> <p>After MDIS is set, even before STF asserts, data read from the channel registers is not meaningful and writes are ineffective, issuing a Bus Error. When the MDIS bit is asserted while the microcode is executing, the eTPU will stop when the thread is complete.</p> <p>Stop completes on the next eTPU clock after the stop condition is valid.</p> <p><b>NOTE:</b> Once MDIS is switched from 1 to 0 or vice versa, it must not write a different value until STF changes accordingly.</p> <p>The MDIS bit is write-protected when VIS=1.</p> <p>See Register Reset Values in the chip-specific eTPU information.</p> <p>0 eTPU Engine runs.</p> <p>1 Commands Engine to stop its clocks.</p> |
| 2<br>Reserved | <p>Reserved.</p> <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 3<br>STF      | <p>Stop Flag Bit.</p> <p>Each Engine asserts its stop flag (STF) to indicate that it has stopped. Only then the host can assume that the Engine has actually stopped. The eTPU system is fully stopped when the STF bits of both eTPU Engines are asserted. In case of a stop request, the eTPU responds by acknowledging stop only after both eTPU 1 and eTPU 2 have been stopped. Engine only stops when any ongoing thread is complete also in this case.</p> <p>Summarizing Engine stop conditions, which STF reflects:</p> <p>STF_1 := (after stop completed) MDIS_1   device stop request.</p> <p>STF_2 := (after stop completed) MDIS_2   device stop request.</p> <p>STF_1 and STF_2 mean STF bit from engine 1 and STF bit from engine 2, respectively.</p> <p>See Register Reset Values in the chip-specific eTPU information.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

*Table continues on the next page...*

**eTPU\_SCR\_ECR\_ENG1 field descriptions (continued)**

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                |                                    |  |  |          |          |     |   |   |     |   |   |     |   |   |     |    |   |     |    |    |     |    |    |     |     |    |     |     |     |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|------------------------------------|--|--|----------|----------|-----|---|---|-----|---|---|-----|---|---|-----|----|---|-----|----|----|-----|----|----|-----|-----|----|-----|-----|-----|
|                  | 0 Engine is operating.<br>1 Engine has stopped (after the local MDIS bit has been asserted, or after the IP-Bus stop line has been asserted).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                |                                    |  |  |          |          |     |   |   |     |   |   |     |   |   |     |    |   |     |    |    |     |    |    |     |     |    |     |     |     |
| 4–7<br>Reserved  | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                |                                    |  |  |          |          |     |   |   |     |   |   |     |   |   |     |    |   |     |    |    |     |    |    |     |     |    |     |     |     |
| 8<br>HLTF        | Halt Mode Flag.<br><br>If eTPU Engine entered halt state, this flag is asserted. The flag remains asserted while the microengine is in halt state, even during a single-step or forced instruction execution. See <a href="#">Development support features</a> for further details about entering Halt Mode.<br><br>0 eTPU Engine is not halted.<br>1 eTPU Engine is halted.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                |                                    |  |  |          |          |     |   |   |     |   |   |     |   |   |     |    |   |     |    |    |     |    |    |     |     |    |     |     |     |
| 9–11<br>Reserved | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                |                                    |  |  |          |          |     |   |   |     |   |   |     |   |   |     |    |   |     |    |    |     |    |    |     |     |    |     |     |     |
| 12<br>FCSS       | Filter Clock Source Selection<br><br>Speeds up the filter clock source before the prescaler, allowing more input capture resolution at minimum prescaling.<br><br><b>NOTE:</b> FCSS=1 also makes the channel work on T2/T4 timing mode (see <a href="#">T2/T4 channel timing</a> ).<br><br>0 Use eTPU clock / 2 as EDF clock source before prescaler.<br>1 Use eTPU clock as EDF clock source before prescaler                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                |                                    |  |  |          |          |     |   |   |     |   |   |     |   |   |     |    |   |     |    |    |     |    |    |     |     |    |     |     |     |
| 13–15<br>FPSCK   | Filter Prescaler Clock Control.<br><br>FPSCK controls the prescaling of the clocks used in digital filters for the channel input signals and TCRCLK input, as shown in the table below. Filtering can be controlled independently by each engine, but all input digital filters in the same engine have same clock prescaling. For more details, see <a href="#">Filter clock prescaler</a> .<br><br>A new value written to FPSCK only becomes effective when the filter prescaler finishes the current count.<br><br><b>Table 35-14. Filter prescaler clock control</b> <table><tr><th>Filter control</th><th colspan="2">Sample on system clock divided by:</th></tr><tr><td></td><th>FCSS = 0</th><th>FCSS = 1</th></tr><tr><td>000</td><td>2</td><td>1</td></tr><tr><td>001</td><td>4</td><td>2</td></tr><tr><td>010</td><td>8</td><td>4</td></tr><tr><td>011</td><td>16</td><td>8</td></tr><tr><td>100</td><td>32</td><td>16</td></tr><tr><td>101</td><td>64</td><td>32</td></tr><tr><td>110</td><td>128</td><td>64</td></tr><tr><td>111</td><td>256</td><td>128</td></tr></table> | Filter control | Sample on system clock divided by: |  |  | FCSS = 0 | FCSS = 1 | 000 | 2 | 1 | 001 | 4 | 2 | 010 | 8 | 4 | 011 | 16 | 8 | 100 | 32 | 16 | 101 | 64 | 32 | 110 | 128 | 64 | 111 | 256 | 128 |
| Filter control   | Sample on system clock divided by:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                |                                    |  |  |          |          |     |   |   |     |   |   |     |   |   |     |    |   |     |    |    |     |    |    |     |     |    |     |     |     |
|                  | FCSS = 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | FCSS = 1       |                                    |  |  |          |          |     |   |   |     |   |   |     |   |   |     |    |   |     |    |    |     |    |    |     |     |    |     |     |     |
| 000              | 2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 1              |                                    |  |  |          |          |     |   |   |     |   |   |     |   |   |     |    |   |     |    |    |     |    |    |     |     |    |     |     |     |
| 001              | 4                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 2              |                                    |  |  |          |          |     |   |   |     |   |   |     |   |   |     |    |   |     |    |    |     |    |    |     |     |    |     |     |     |
| 010              | 8                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 4              |                                    |  |  |          |          |     |   |   |     |   |   |     |   |   |     |    |   |     |    |    |     |    |    |     |     |    |     |     |     |
| 011              | 16                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 8              |                                    |  |  |          |          |     |   |   |     |   |   |     |   |   |     |    |   |     |    |    |     |    |    |     |     |    |     |     |     |
| 100              | 32                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 16             |                                    |  |  |          |          |     |   |   |     |   |   |     |   |   |     |    |   |     |    |    |     |    |    |     |     |    |     |     |     |
| 101              | 64                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 32             |                                    |  |  |          |          |     |   |   |     |   |   |     |   |   |     |    |   |     |    |    |     |    |    |     |     |    |     |     |     |
| 110              | 128                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 64             |                                    |  |  |          |          |     |   |   |     |   |   |     |   |   |     |    |   |     |    |    |     |    |    |     |     |    |     |     |     |
| 111              | 256                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 128            |                                    |  |  |          |          |     |   |   |     |   |   |     |   |   |     |    |   |     |    |    |     |    |    |     |     |    |     |     |     |

Table continues on the next page...

**eTPU\_SCR\_ECR\_ENG1 field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |      |                         |    |                                                                                                                                                                                                                                                                                              |    |                                                                                                                          |    |                                                                                                                                                                 |    |                                                                                                                                                                                                                                                                                                                                                                                 |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|-------------------------|----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|--------------------------------------------------------------------------------------------------------------------------|----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 16–17<br>CDFC     | <p>Channel Digital Filter Control.</p> <p>These bits select a digital filtering mode for the channels when configured as inputs for improved noise immunity (refer to the table below). The eTPU has three digital filtering modes for the channels which provide programmable trade-off between signal latency and noise immunity (see <a href="#">Enhanced Digital Filter - EDF</a>). Changing CDFC during eTPU normal input channel operation is not recommended since it changes the behavior of the transition detection logic while executing its operation.</p> <p><b>Table 35-13. Channel digital filter control</b></p> <table> <tr> <th>CDFC</th><th>Selected digital filter</th></tr> <tr> <td>00</td><td>TPU2/3 Two Sample Mode: Using the filter clock which is the system clock divided by (2, 4, 8,..., 256) as a sampling clock (selected by <a href="#">FPSCK</a>), comparing two consecutive samples which agree with each other sets the input signal state. This is the default reset state.</td></tr> <tr> <td>01</td><td>eTPU bypass mode: the input signal is taken unfiltered, also making the channels work on T2/T4 timing mode.<sup>1</sup></td></tr> <tr> <td>10</td><td>eTPU Three Sample Mode: Similar to the TPU2/3 two sample mode, but comparing three consecutive samples which agree with each other sets the input signal state.</td></tr> <tr> <td>11</td><td>eTPU Continuous Mode: Signal need to be stable for the whole filter clock period. This mode compares all the values at the rate of eTPU clock (FCSS=1) or eTPU clock divided by two(FCSS=0), between two consecutive filter clock pulses. Signal needs to be continuously stable for the entire period. If all the values agree with each other, input signal state is updated.</td></tr> </table> <p>1. See <a href="#">T2/T4 channel timing</a>.</p> | CDFC | Selected digital filter | 00 | TPU2/3 Two Sample Mode: Using the filter clock which is the system clock divided by (2, 4, 8,..., 256) as a sampling clock (selected by <a href="#">FPSCK</a> ), comparing two consecutive samples which agree with each other sets the input signal state. This is the default reset state. | 01 | eTPU bypass mode: the input signal is taken unfiltered, also making the channels work on T2/T4 timing mode. <sup>1</sup> | 10 | eTPU Three Sample Mode: Similar to the TPU2/3 two sample mode, but comparing three consecutive samples which agree with each other sets the input signal state. | 11 | eTPU Continuous Mode: Signal need to be stable for the whole filter clock period. This mode compares all the values at the rate of eTPU clock (FCSS=1) or eTPU clock divided by two(FCSS=0), between two consecutive filter clock pulses. Signal needs to be continuously stable for the entire period. If all the values agree with each other, input signal state is updated. |
| CDFC              | Selected digital filter                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |      |                         |    |                                                                                                                                                                                                                                                                                              |    |                                                                                                                          |    |                                                                                                                                                                 |    |                                                                                                                                                                                                                                                                                                                                                                                 |
| 00                | TPU2/3 Two Sample Mode: Using the filter clock which is the system clock divided by (2, 4, 8,..., 256) as a sampling clock (selected by <a href="#">FPSCK</a> ), comparing two consecutive samples which agree with each other sets the input signal state. This is the default reset state.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |      |                         |    |                                                                                                                                                                                                                                                                                              |    |                                                                                                                          |    |                                                                                                                                                                 |    |                                                                                                                                                                                                                                                                                                                                                                                 |
| 01                | eTPU bypass mode: the input signal is taken unfiltered, also making the channels work on T2/T4 timing mode. <sup>1</sup>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |      |                         |    |                                                                                                                                                                                                                                                                                              |    |                                                                                                                          |    |                                                                                                                                                                 |    |                                                                                                                                                                                                                                                                                                                                                                                 |
| 10                | eTPU Three Sample Mode: Similar to the TPU2/3 two sample mode, but comparing three consecutive samples which agree with each other sets the input signal state.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |      |                         |    |                                                                                                                                                                                                                                                                                              |    |                                                                                                                          |    |                                                                                                                                                                 |    |                                                                                                                                                                                                                                                                                                                                                                                 |
| 11                | eTPU Continuous Mode: Signal need to be stable for the whole filter clock period. This mode compares all the values at the rate of eTPU clock (FCSS=1) or eTPU clock divided by two(FCSS=0), between two consecutive filter clock pulses. Signal needs to be continuously stable for the entire period. If all the values agree with each other, input signal state is updated.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |      |                         |    |                                                                                                                                                                                                                                                                                              |    |                                                                                                                          |    |                                                                                                                                                                 |    |                                                                                                                                                                                                                                                                                                                                                                                 |
| 18<br>Reserved    | <p>Reserved.</p> <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |      |                         |    |                                                                                                                                                                                                                                                                                              |    |                                                                                                                          |    |                                                                                                                                                                 |    |                                                                                                                                                                                                                                                                                                                                                                                 |
| 19–23<br>ERBA     | <p>Engine Relative Base Address.</p> <p>This field value is concatenated with the AID instruction field in engine relative address mode to form the SDM address (see <a href="#">Engine Relative Addressing Mode</a>).</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |      |                         |    |                                                                                                                                                                                                                                                                                              |    |                                                                                                                          |    |                                                                                                                                                                 |    |                                                                                                                                                                                                                                                                                                                                                                                 |
| 24<br>SPPDIS      | <p>Schedule Priority Passing Disable.</p> <p>SPPDIS is used to disable the priority passing mechanism of the microengine scheduler (see <a href="#">Primary Scheme - Priority Among Channels on Different Levels</a>).</p> <p><b>NOTE:</b> SPPDIS bit must not be changed while any channel is enabled.</p> <p>0 Scheduler priority passing mechanism enabled.</p> <p>1 Scheduler priority passing mechanism disabled.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |      |                         |    |                                                                                                                                                                                                                                                                                              |    |                                                                                                                          |    |                                                                                                                                                                 |    |                                                                                                                                                                                                                                                                                                                                                                                 |
| 25–26<br>Reserved | <p>Reserved.</p> <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |      |                         |    |                                                                                                                                                                                                                                                                                              |    |                                                                                                                          |    |                                                                                                                                                                 |    |                                                                                                                                                                                                                                                                                                                                                                                 |
| 27–31<br>ETB      | Entry Table Base.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |      |                         |    |                                                                                                                                                                                                                                                                                              |    |                                                                                                                          |    |                                                                                                                                                                 |    |                                                                                                                                                                                                                                                                                                                                                                                 |

*Table continues on the next page...*

**eTPU\_SCR\_ECR\_ENG1 field descriptions (continued)**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                             |                                                              |                                                              |       |       |       |       |       |       |       |        |       |       |       |       |       |        |        |       |        |        |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------|--------------------------------------------------------------|--------------------------------------------------------------|-------|-------|-------|-------|-------|-------|-------|--------|-------|-------|-------|-------|-------|--------|--------|-------|--------|--------|
|       | The field determines the location of the microcode entry table for the eTPU functions in SCM (see <a href="#">Entry Points</a> ). The following table shows the entry table base address options.                                                                                                                                                                                                                                                                                                    |                                                             |                                                              |                                                              |       |       |       |       |       |       |       |        |       |       |       |       |       |        |        |       |        |        |
|       | <b>Table 35-12. Entry table base address options</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                                                             |                                                              |                                                              |       |       |       |       |       |       |       |        |       |       |       |       |       |        |        |       |        |        |
|       | <table><tr><th>ETB</th><th>Entry Table Base Address for CPU Host Address (byte format)</th><th>Entry Table Base Address for Microcode Address (word format)</th></tr><tr><td>00000</td><td>0x000</td><td>0x000</td></tr><tr><td>00001</td><td>0x800</td><td>0x200</td></tr><tr><td>00010</td><td>0x1000</td><td>0x400</td></tr><tr><td>-----</td><td>-----</td><td>-----</td></tr><tr><td>11110</td><td>0xF000</td><td>0x3C00</td></tr><tr><td>11111</td><td>0xF800</td><td>0x3E00</td></tr></table> | ETB                                                         | Entry Table Base Address for CPU Host Address (byte format)  | Entry Table Base Address for Microcode Address (word format) | 00000 | 0x000 | 0x000 | 00001 | 0x800 | 0x200 | 00010 | 0x1000 | 0x400 | ----- | ----- | ----- | 11110 | 0xF000 | 0x3C00 | 11111 | 0xF800 | 0x3E00 |
|       | ETB                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Entry Table Base Address for CPU Host Address (byte format) | Entry Table Base Address for Microcode Address (word format) |                                                              |       |       |       |       |       |       |       |        |       |       |       |       |       |        |        |       |        |        |
|       | 00000                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 0x000                                                       | 0x000                                                        |                                                              |       |       |       |       |       |       |       |        |       |       |       |       |       |        |        |       |        |        |
|       | 00001                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 0x800                                                       | 0x200                                                        |                                                              |       |       |       |       |       |       |       |        |       |       |       |       |       |        |        |       |        |        |
|       | 00010                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 0x1000                                                      | 0x400                                                        |                                                              |       |       |       |       |       |       |       |        |       |       |       |       |       |        |        |       |        |        |
|       | -----                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | -----                                                       | -----                                                        |                                                              |       |       |       |       |       |       |       |        |       |       |       |       |       |        |        |       |        |        |
|       | 11110                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 0xF000                                                      | 0x3C00                                                       |                                                              |       |       |       |       |       |       |       |        |       |       |       |       |       |        |        |       |        |        |
| 11111 | 0xF800                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 0x3E00                                                      |                                                              |                                                              |       |       |       |       |       |       |       |        |       |       |       |       |       |        |        |       |        |        |
|       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                             |                                                              |                                                              |       |       |       |       |       |       |       |        |       |       |       |       |       |        |        |       |        |        |
|       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                             |                                                              |                                                              |       |       |       |       |       |       |       |        |       |       |       |       |       |        |        |       |        |        |
|       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                             |                                                              |                                                              |       |       |       |       |       |       |       |        |       |       |       |       |       |        |        |       |        |        |
|       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                             |                                                              |                                                              |       |       |       |       |       |       |       |        |       |       |       |       |       |        |        |       |        |        |
|       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                             |                                                              |                                                              |       |       |       |       |       |       |       |        |       |       |       |       |       |        |        |       |        |        |

3. See [T2/T4 channel timing](#).

### 35.7.6 eTPU Engine 2 Configuration Register (eTPU\_SCR\_ECR\_ENG2)

This register holds configuration and status fields for engine 2.

#### NOTE

See Register Reset Values in the chip-specific eTPU information.

Address: 0h base + 18h offset = 18h

| Bit   | 0 | 1 | 2 | 3   | 4 | 5 | 6 | 7 | 8    | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|-------|---|---|---|-----|---|---|---|---|------|---|----|----|----|----|----|----|
| R     |   |   | 0 | STF |   |   | 0 |   | HLTF |   | 0  |    |    |    |    |    |
| W     |   |   |   |     |   |   |   |   |      |   |    |    |    |    |    |    |
| Reset | 0 | * | 0 | 0   | 0 | 0 | 0 | 0 | *    | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

## System Configuration memory map and registers

| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| R     |    |    | 0  |    |    |    |    |    |    | 0  |    |    |    |    |    |    |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

### \* Notes:

- HLTF field: HLTF field reset value is 1 if, after a reset, the eTPU Nexus debug support is on and a breakpoint request is asserted. Otherwise the reset value is 0.
- MDIS field: Reset value is 0.

## eTPU\_SCR\_ECR\_ENG2 field descriptions

| Field     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>FEND | <p>Force End.</p> <p>FEND assertion terminates any current running thread as if an END instruction have been executed (see <a href="#">Ending Current Thread - END</a>).</p> <p>This bit is self-negating when the thread ends.</p> <p><b>NOTE:</b> Only on rare occasions (e.g., during a long stall, see <a href="#">Microengine Stall</a>) FEND can be read as 1, because it negates as soon as the end begins execution.</p> <p>Writing FEND = 1 is ignored and FEND stays 0 when the microengine is in TST, halted, stopped, or idle (no thread executing).</p> <p>0 Normal operation.</p> <p>1 Ends any ongoing thread.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 1<br>MDIS | <p>Module Disable Bit.</p> <p>When MDIS is set, the Engine shuts down its internal clocks, going into Module Disable Mode. TCR1 and TCR2 cease to increment, and input sampling stops. The Engine asserts the stop flag (STF) bit to indicate that it has stopped. However, the BIU continues to run, and the Host can access all registers except for the channel registers (see channel registers starting with eTPU Channel Configuration-Control registers)</p> <p><b>NOTE:</b> The Timebase registers can still be read with MDIS=1, but writes are ineffective and a Bus Error is issued. Global Channel Registers and SDM can be accessed normally.</p> <p>After MDIS is set, even before STF asserts, data read from the channel registers is not meaningful and writes are ineffective, issuing a Bus Error. When the MDIS bit is asserted while the microcode is executing, the eTPU will stop when the thread is complete.</p> <p>Stop completes on the next eTPU clock after the stop condition is valid.</p> <p><b>NOTE:</b> Once MDIS is switched from 1 to 0 or vice versa, it must not write a different value until STF changes accordingly.</p> <p>The MDIS bit is write-protected when VIS=1.</p> <p>See Register Reset Values in the chip-specific eTPU information.</p> |

Table continues on the next page...



**eTPU\_SCR\_ECR\_ENG2 field descriptions (continued)**

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                  | <p>0 eTPU Engine runs.</p> <p>1 Commands Engine to stop its clocks.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 2<br>Reserved    | <p>Reserved.</p> <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 3<br>STF         | <p>Stop Flag Bit.</p> <p>Each Engine asserts its stop flag (STF) to indicate that it has stopped. Only then the host can assume that the Engine has actually stopped. The eTPU system is fully stopped when the STF bits of both eTPU Engines are asserted. In case of IP-Bus stop, the eTPU system responds by acknowledging stop only after both eTPU 1 and eTPU 2 have been stopped. Engine only stops when any ongoing thread is complete also in this case.</p> <p>Summarizing Engine stop conditions, which STF reflects:</p> <p>STF_1 := (after stop completed) MDIS_1   device stop request.</p> <p>STF_2 := (after stop completed) MDIS_2   device stop request.</p> <p>STF_1 and STF_2 mean STF bit from engine 1 and STF bit from engine 2, respectively.</p> <p>See Register Reset Values in the chip-specific eTPU information.</p> <p>0 Engine is operating.</p> <p>1 Engine has stopped (after the local MDIS bit has been asserted, or after the IP-Bus stop line has been asserted).</p> |
| 4–7<br>Reserved  | <p>Reserved.</p> <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 8<br>HLTF        | <p>Halt Mode Flag.</p> <p>If eTPU Engine entered halt state, this flag is asserted. The flag remains asserted while the microengine is in halt state, even during a single-step or forced instruction execution. See <a href="#">Development support features</a> for further details about entering Halt Mode.</p> <p>0 eTPU Engine is not halted.</p> <p>1 eTPU Engine is halted.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 9–11<br>Reserved | <p>Reserved.</p> <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 12<br>FCSS       | <p>Filter Clock Source Selection</p> <p>Speeds up the filter clock source before the prescaler, allowing more input capture resolution at minimum prescaling.</p> <p><b>NOTE:</b> FCSS=1 also makes the channel work on T2/T4 timing mode (see <a href="#">T2/T4 channel timing</a>).</p> <p>0 Use eTPU clock / 2 as EDF clock source before prescaler.</p> <p>1 Use eTPU clock as EDF clock source before prescaler</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 13–15<br>FPSCK   | <p>Filter Prescaler Clock Control.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

*Table continues on the next page...*

**eTPU\_SCR\_ECR\_ENG2 field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                |                                    |     |                                                                                                                                                                                                                                                                                               |     |                                                                                                                          |     |                                                                                                                                                                 |     |                                                                                                                                                                                                                                                                                                                                                                                 |     |    |     |    |     |     |     |     |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|------------------------------------|-----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|--------------------------------------------------------------------------------------------------------------------------|-----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|----|-----|----|-----|-----|-----|-----|
|                | <p>FPSCCK controls the prescaling of the clocks used in digital filters for the channel input signals and TCRCLK input, as shown in the table below. Filtering can be controlled independently by each engine, but all input digital filters in the same engine have same clock prescaling. For more details, see <a href="#">Filter clock prescaler</a>.</p> <p>A new value written to FPSCCK only becomes effective when the filter prescaler finishes the current count.</p> <p><b>Table 35-18. Filter prescaler clock control</b></p> <table> <tr> <th>Filter control</th><th>Sample on system clock divided by:</th></tr> <tr> <td>000</td><td>2</td></tr> <tr> <td>001</td><td>4</td></tr> <tr> <td>010</td><td>8</td></tr> <tr> <td>011</td><td>16</td></tr> <tr> <td>100</td><td>32</td></tr> <tr> <td>101</td><td>64</td></tr> <tr> <td>110</td><td>128</td></tr> <tr> <td>111</td><td>256</td></tr> </table>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Filter control | Sample on system clock divided by: | 000 | 2                                                                                                                                                                                                                                                                                             | 001 | 4                                                                                                                        | 010 | 8                                                                                                                                                               | 011 | 16                                                                                                                                                                                                                                                                                                                                                                              | 100 | 32 | 101 | 64 | 110 | 128 | 111 | 256 |
| Filter control | Sample on system clock divided by:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                |                                    |     |                                                                                                                                                                                                                                                                                               |     |                                                                                                                          |     |                                                                                                                                                                 |     |                                                                                                                                                                                                                                                                                                                                                                                 |     |    |     |    |     |     |     |     |
| 000            | 2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                |                                    |     |                                                                                                                                                                                                                                                                                               |     |                                                                                                                          |     |                                                                                                                                                                 |     |                                                                                                                                                                                                                                                                                                                                                                                 |     |    |     |    |     |     |     |     |
| 001            | 4                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                |                                    |     |                                                                                                                                                                                                                                                                                               |     |                                                                                                                          |     |                                                                                                                                                                 |     |                                                                                                                                                                                                                                                                                                                                                                                 |     |    |     |    |     |     |     |     |
| 010            | 8                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                |                                    |     |                                                                                                                                                                                                                                                                                               |     |                                                                                                                          |     |                                                                                                                                                                 |     |                                                                                                                                                                                                                                                                                                                                                                                 |     |    |     |    |     |     |     |     |
| 011            | 16                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                |                                    |     |                                                                                                                                                                                                                                                                                               |     |                                                                                                                          |     |                                                                                                                                                                 |     |                                                                                                                                                                                                                                                                                                                                                                                 |     |    |     |    |     |     |     |     |
| 100            | 32                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                |                                    |     |                                                                                                                                                                                                                                                                                               |     |                                                                                                                          |     |                                                                                                                                                                 |     |                                                                                                                                                                                                                                                                                                                                                                                 |     |    |     |    |     |     |     |     |
| 101            | 64                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                |                                    |     |                                                                                                                                                                                                                                                                                               |     |                                                                                                                          |     |                                                                                                                                                                 |     |                                                                                                                                                                                                                                                                                                                                                                                 |     |    |     |    |     |     |     |     |
| 110            | 128                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                |                                    |     |                                                                                                                                                                                                                                                                                               |     |                                                                                                                          |     |                                                                                                                                                                 |     |                                                                                                                                                                                                                                                                                                                                                                                 |     |    |     |    |     |     |     |     |
| 111            | 256                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                |                                    |     |                                                                                                                                                                                                                                                                                               |     |                                                                                                                          |     |                                                                                                                                                                 |     |                                                                                                                                                                                                                                                                                                                                                                                 |     |    |     |    |     |     |     |     |
| 16–17<br>CDFC  | <p>Channel Digital Filter Control.</p> <p>These bits select a digital filtering mode for the channels when configured as inputs for improved noise immunity (refer to the table below). The eTPU has three digital filtering modes for the channels which provide programmable trade-off between signal latency and noise immunity (see <a href="#">Enhanced Digital Filter - EDF</a>). Changing CDFC during eTPU normal input channel operation is not recommended since it changes the behavior of the transition detection logic while executing its operation.</p> <p><b>Table 35-17. Channel digital filter control</b></p> <table> <tr> <th>CDFC</th><th>Selected digital filter</th></tr> <tr> <td>00</td><td>TPU2/3 Two Sample Mode: Using the filter clock which is the system clock divided by (2, 4, 8,..., 256) as a sampling clock (selected by <a href="#">FPSCCK</a>), comparing two consecutive samples which agree with each other sets the input signal state. This is the default reset state.</td></tr> <tr> <td>01</td><td>eTPU bypass mode: the input signal is taken unfiltered, also making the channels work on T2/T4 timing mode.<sup>1</sup></td></tr> <tr> <td>10</td><td>eTPU Three Sample Mode: Similar to the TPU2/3 two sample mode, but comparing three consecutive samples which agree with each other sets the input signal state.</td></tr> <tr> <td>11</td><td>eTPU Continuous Mode: Signal need to be stable for the whole filter clock period. This mode compares all the values at the rate of eTPU clock (FCSS=1) or eTPU clock divided by two(FCSS=0), between two consecutive filter clock pulses. Signal needs to be continuously stable for the entire period. If all the values agree with each other, input signal state is updated.</td></tr> </table> <p>1. See <a href="#">T2/T4 channel timing</a>.</p> | CDFC           | Selected digital filter            | 00  | TPU2/3 Two Sample Mode: Using the filter clock which is the system clock divided by (2, 4, 8,..., 256) as a sampling clock (selected by <a href="#">FPSCCK</a> ), comparing two consecutive samples which agree with each other sets the input signal state. This is the default reset state. | 01  | eTPU bypass mode: the input signal is taken unfiltered, also making the channels work on T2/T4 timing mode. <sup>1</sup> | 10  | eTPU Three Sample Mode: Similar to the TPU2/3 two sample mode, but comparing three consecutive samples which agree with each other sets the input signal state. | 11  | eTPU Continuous Mode: Signal need to be stable for the whole filter clock period. This mode compares all the values at the rate of eTPU clock (FCSS=1) or eTPU clock divided by two(FCSS=0), between two consecutive filter clock pulses. Signal needs to be continuously stable for the entire period. If all the values agree with each other, input signal state is updated. |     |    |     |    |     |     |     |     |
| CDFC           | Selected digital filter                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                |                                    |     |                                                                                                                                                                                                                                                                                               |     |                                                                                                                          |     |                                                                                                                                                                 |     |                                                                                                                                                                                                                                                                                                                                                                                 |     |    |     |    |     |     |     |     |
| 00             | TPU2/3 Two Sample Mode: Using the filter clock which is the system clock divided by (2, 4, 8,..., 256) as a sampling clock (selected by <a href="#">FPSCCK</a> ), comparing two consecutive samples which agree with each other sets the input signal state. This is the default reset state.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                |                                    |     |                                                                                                                                                                                                                                                                                               |     |                                                                                                                          |     |                                                                                                                                                                 |     |                                                                                                                                                                                                                                                                                                                                                                                 |     |    |     |    |     |     |     |     |
| 01             | eTPU bypass mode: the input signal is taken unfiltered, also making the channels work on T2/T4 timing mode. <sup>1</sup>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                |                                    |     |                                                                                                                                                                                                                                                                                               |     |                                                                                                                          |     |                                                                                                                                                                 |     |                                                                                                                                                                                                                                                                                                                                                                                 |     |    |     |    |     |     |     |     |
| 10             | eTPU Three Sample Mode: Similar to the TPU2/3 two sample mode, but comparing three consecutive samples which agree with each other sets the input signal state.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                |                                    |     |                                                                                                                                                                                                                                                                                               |     |                                                                                                                          |     |                                                                                                                                                                 |     |                                                                                                                                                                                                                                                                                                                                                                                 |     |    |     |    |     |     |     |     |
| 11             | eTPU Continuous Mode: Signal need to be stable for the whole filter clock period. This mode compares all the values at the rate of eTPU clock (FCSS=1) or eTPU clock divided by two(FCSS=0), between two consecutive filter clock pulses. Signal needs to be continuously stable for the entire period. If all the values agree with each other, input signal state is updated.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                |                                    |     |                                                                                                                                                                                                                                                                                               |     |                                                                                                                          |     |                                                                                                                                                                 |     |                                                                                                                                                                                                                                                                                                                                                                                 |     |    |     |    |     |     |     |     |
| 18<br>Reserved | <p>Reserved.</p> <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                |                                    |     |                                                                                                                                                                                                                                                                                               |     |                                                                                                                          |     |                                                                                                                                                                 |     |                                                                                                                                                                                                                                                                                                                                                                                 |     |    |     |    |     |     |     |     |

Table continues on the next page...

**eTPU\_SCR\_ECR\_ENG2 field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                                                              |                                                             |                                                              |       |       |       |       |       |       |       |        |       |       |       |       |       |        |        |       |        |        |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------|-------------------------------------------------------------|--------------------------------------------------------------|-------|-------|-------|-------|-------|-------|-------|--------|-------|-------|-------|-------|-------|--------|--------|-------|--------|--------|
| 19–23<br>ERBA     | Engine Relative Base Address.<br><br>This field value is concatenated with the AID instruction field in engine relative address mode to form the SDM address (see <a href="#">Engine Relative Addressing Mode</a> ).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                                                              |                                                             |                                                              |       |       |       |       |       |       |       |        |       |       |       |       |       |        |        |       |        |        |
| 24<br>SPPDIS      | Schedule Priority Passing Disable.<br><br>SPPDIS is used to disable the priority passing mechanism of the microengine scheduler (see <a href="#">Primary Scheme - Priority Among Channels on Different Levels</a> ).<br><br><b>NOTE:</b> SPPDIS bit must not be changed while any channel is enabled.<br><br>0 Scheduler priority passing mechanism enabled.<br>1 Scheduler priority passing mechanism disabled.                                                                                                                                                                                                                                                                                                                                                                            |                                                              |                                                             |                                                              |       |       |       |       |       |       |       |        |       |       |       |       |       |        |        |       |        |        |
| 25–26<br>Reserved | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                                              |                                                             |                                                              |       |       |       |       |       |       |       |        |       |       |       |       |       |        |        |       |        |        |
| 27–31<br>ETB      | Entry Table Base.<br><br>The field determines the location of the microcode entry table for the eTPU functions in SCM (see <a href="#">Entry Points</a> ). The following table shows the entry table base address options.<br><br><b>Table 35-16. Entry table base address options</b> <table><tr><th>ETB</th><th>Entry Table Base Address for CPU Host Address (byte format)</th><th>Entry Table Base Address for Microcode Address (word format)</th></tr><tr><td>00000</td><td>0x000</td><td>0x000</td></tr><tr><td>00001</td><td>0x800</td><td>0x200</td></tr><tr><td>00010</td><td>0x1000</td><td>0x400</td></tr><tr><td>-----</td><td>-----</td><td>-----</td></tr><tr><td>11110</td><td>0xF000</td><td>0x3C00</td></tr><tr><td>11111</td><td>0xF800</td><td>0x3E00</td></tr></table> | ETB                                                          | Entry Table Base Address for CPU Host Address (byte format) | Entry Table Base Address for Microcode Address (word format) | 00000 | 0x000 | 0x000 | 00001 | 0x800 | 0x200 | 00010 | 0x1000 | 0x400 | ----- | ----- | ----- | 11110 | 0xF000 | 0x3C00 | 11111 | 0xF800 | 0x3E00 |
| ETB               | Entry Table Base Address for CPU Host Address (byte format)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Entry Table Base Address for Microcode Address (word format) |                                                             |                                                              |       |       |       |       |       |       |       |        |       |       |       |       |       |        |        |       |        |        |
| 00000             | 0x000                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 0x000                                                        |                                                             |                                                              |       |       |       |       |       |       |       |        |       |       |       |       |       |        |        |       |        |        |
| 00001             | 0x800                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 0x200                                                        |                                                             |                                                              |       |       |       |       |       |       |       |        |       |       |       |       |       |        |        |       |        |        |
| 00010             | 0x1000                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 0x400                                                        |                                                             |                                                              |       |       |       |       |       |       |       |        |       |       |       |       |       |        |        |       |        |        |
| -----             | -----                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | -----                                                        |                                                             |                                                              |       |       |       |       |       |       |       |        |       |       |       |       |       |        |        |       |        |        |
| 11110             | 0xF000                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 0x3C00                                                       |                                                             |                                                              |       |       |       |       |       |       |       |        |       |       |       |       |       |        |        |       |        |        |
| 11111             | 0xF800                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 0x3E00                                                       |                                                             |                                                              |       |       |       |       |       |       |       |        |       |       |       |       |       |        |        |       |        |        |

3. See [T2/T4 channel timing](#).

## 35.8 Timebase memory map and registers

Time Base registers allows configuration and visibility of internally-generated time bases TCR1 and TCR2. There is one of each of these registers for each eTPU Engine.

### NOTE

Writes to these registers issue bus error and are ineffective when MDIS=1. Reads are always allowed.

## eTPU\_TBR memory map

| Address offset (hex) | Register name                                                             | Width (in bits) | Access | Reset value | Section/ page               |
|----------------------|---------------------------------------------------------------------------|-----------------|--------|-------------|-----------------------------|
| 20                   | eTPU Engine 1 Time Base Configuration Register (eTPU_TBR_TBCR_ENG1)       | 32              | R/W    | 2000_0000h  | <a href="#">35.8.1/1320</a> |
| 24                   | eTPU Engine 1 Time Base 1 (TCR1) Visibility Register (eTPU_TBR_TB1R_ENG1) | 32              | R/W    | 0000_0000h  | <a href="#">35.8.2/1324</a> |
| 28                   | eTPU Engine 1 Time Base 2 (TCR2) Visibility Register (eTPU_TBR_TB2R_ENG1) | 32              | R/W    | 0000_0000h  | <a href="#">35.8.3/1324</a> |
| 2C                   | eTPU Engine 1 STAC Configuration Register (eTPU_TBR_STAC_ENG1)            | 32              | R/W    | See section | <a href="#">35.8.4/1325</a> |
| 40                   | eTPU Engine 2 Time Base Configuration Register (eTPU_TBR_TBCR_ENG2)       | 32              | R/W    | 2000_0000h  | <a href="#">35.8.5/1327</a> |
| 44                   | eTPU Engine 2 Time Base 1 (TCR1) Visibility Register (eTPU_TBR_TB1R_ENG2) | 32              | R/W    | 0000_0000h  | <a href="#">35.8.6/1331</a> |
| 48                   | eTPU Engine 2 Time Base 2 (TCR2) Visibility Register (eTPU_TBR_TB2R_ENG2) | 32              | R/W    | 0000_0000h  | <a href="#">35.8.7/1331</a> |
| 4C                   | eTPU Engine 2 STAC Configuration Register (eTPU_TBR_STAC_ENG2)            | 32              | R/W    | See section | <a href="#">35.8.8/1332</a> |

### 35.8.1 eTPU Engine 1 Time Base Configuration Register (eTPU\_TBR\_TBCR\_ENG1)

This register configures several timebase options.

Address: 0h base + 20h offset = 20h

|       |         |    |        |       |    |    |    |    |       |    |       |    |    |    |    |    |
|-------|---------|----|--------|-------|----|----|----|----|-------|----|-------|----|----|----|----|----|
| Bit   | 0       | 1  | 2      | 3     | 4  | 5  | 6  | 7  | 8     | 9  | 10    | 11 | 12 | 13 | 14 | 15 |
| R     | TCR2CTL |    |        | TCRCF |    |    | AM |    | ATGC  |    | TCR2P |    |    |    |    |    |
| W     |         |    |        |       |    |    |    |    |       |    |       |    |    |    |    |    |
| Reset | 0       | 0  | 1      | 0     | 0  | 0  | 0  | 0  | 0     | 0  | 0     | 0  | 0  | 0  | 0  | 0  |
| Bit   | 16      | 17 | 18     | 19    | 20 | 21 | 22 | 23 | 24    | 25 | 26    | 27 | 28 | 29 | 30 | 31 |
| R     | TCR1CTL |    | TCR1CS | 0     |    |    |    |    | TCR1P |    |       |    |    |    |    |    |
| W     |         |    |        |       |    |    |    |    |       |    |       |    |    |    |    |    |
| Reset | 0       | 0  | 0      | 0     | 0  | 0  | 0  | 0  | 0     | 0  | 0     | 0  | 0  | 0  | 0  | 0  |

#### eTPU\_TBR\_TBCR\_ENG1 field descriptions

| Field          | Description              |
|----------------|--------------------------|
| 0–2<br>TCR2CTL | TCR2 Clock/Gate Control. |

Table continues on the next page...

**eTPU\_TBR\_TBCR\_ENG1 field descriptions (continued)**

| Field        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                                 |                                       |                                 |     |                                                                                                                                                                                                                                                                                                 |                        |     |                                                                 |             |     |                                                                 |              |     |                                                                         |            |     |                             |                        |     |                      |                        |     |                      |                      |     |                                    |                        |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------|---------------------------------------|---------------------------------|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|-----|-----------------------------------------------------------------|-------------|-----|-----------------------------------------------------------------|--------------|-----|-------------------------------------------------------------------------|------------|-----|-----------------------------|------------------------|-----|----------------------|------------------------|-----|----------------------|----------------------|-----|------------------------------------|------------------------|
|              | <p>These bits are part of the TCR2 clocking system (see <a href="#">Time bases</a>). They determine the clock source for TCR2 before the prescaler. TCR2 can count on any detected edge of the TCRCLK signal or use it for gating eTPU clock divided by 8. After reset - TCRCLK signal rising edge is selected. TCR2 can also be clocked by the eTPU clock divided by 8. TCR2CTL also determines the TCRCLK edge selected for angle tooth detection in angle mode. See the following table.</p> <p><b>Table 35-24. TCR2 Clock Source</b></p> <table><tr><th>TCR2CTL</th><th>AM = 0<br/>TCR2 Clock before prescaler</th><th>AM &gt; 0<br/>Angle Tooth detection</th></tr><tr><td>000</td><td>Gated DIV8 clock (eTPU clock / 8). In this case, when the external TCRCLK signal is low, the DIV8 clock is blocked, preventing it from incrementing the TCR2 prescaler. When the external TCRCLK signal is high, TCR2 prescaler is incremented at the frequency of the eTPU clock divided by 8.</td><td>Do not use with AM &gt; 0</td></tr><tr><td>001</td><td>Rise transition on TCRCLK signal increments the TCR2 prescaler.</td><td>Rising edge</td></tr><tr><td>010</td><td>Fall transition on TCRCLK signal increments the TCR2 prescaler.</td><td>Falling edge</td></tr><tr><td>011</td><td>Rise or Fall transition on TCRCLK signal increments the TCR2 prescaler.</td><td>Both edges</td></tr><tr><td>100</td><td>DIV8 clock (eTPU clock / 8)</td><td>Do not use with AM &gt; 0</td></tr><tr><td>101</td><td>Do not use with AM=0</td><td>Do not use with AM &gt; 0</td></tr><tr><td>110</td><td>Do not use with AM=0</td><td>No edge<sup>1</sup></td></tr><tr><td>111</td><td>TCR2 frozen, except as STAC client</td><td>Do not use with AM &gt; 0</td></tr></table> <p>1. TCRCLK edges are not detected by the EAC logic, but they can still be detected by the channel 0 logic if AM=01.</p> | TCR2CTL                         | AM = 0<br>TCR2 Clock before prescaler | AM > 0<br>Angle Tooth detection | 000 | Gated DIV8 clock (eTPU clock / 8). In this case, when the external TCRCLK signal is low, the DIV8 clock is blocked, preventing it from incrementing the TCR2 prescaler. When the external TCRCLK signal is high, TCR2 prescaler is incremented at the frequency of the eTPU clock divided by 8. | Do not use with AM > 0 | 001 | Rise transition on TCRCLK signal increments the TCR2 prescaler. | Rising edge | 010 | Fall transition on TCRCLK signal increments the TCR2 prescaler. | Falling edge | 011 | Rise or Fall transition on TCRCLK signal increments the TCR2 prescaler. | Both edges | 100 | DIV8 clock (eTPU clock / 8) | Do not use with AM > 0 | 101 | Do not use with AM=0 | Do not use with AM > 0 | 110 | Do not use with AM=0 | No edge <sup>1</sup> | 111 | TCR2 frozen, except as STAC client | Do not use with AM > 0 |
| TCR2CTL      | AM = 0<br>TCR2 Clock before prescaler                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | AM > 0<br>Angle Tooth detection |                                       |                                 |     |                                                                                                                                                                                                                                                                                                 |                        |     |                                                                 |             |     |                                                                 |              |     |                                                                         |            |     |                             |                        |     |                      |                        |     |                      |                      |     |                                    |                        |
| 000          | Gated DIV8 clock (eTPU clock / 8). In this case, when the external TCRCLK signal is low, the DIV8 clock is blocked, preventing it from incrementing the TCR2 prescaler. When the external TCRCLK signal is high, TCR2 prescaler is incremented at the frequency of the eTPU clock divided by 8.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Do not use with AM > 0          |                                       |                                 |     |                                                                                                                                                                                                                                                                                                 |                        |     |                                                                 |             |     |                                                                 |              |     |                                                                         |            |     |                             |                        |     |                      |                        |     |                      |                      |     |                                    |                        |
| 001          | Rise transition on TCRCLK signal increments the TCR2 prescaler.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Rising edge                     |                                       |                                 |     |                                                                                                                                                                                                                                                                                                 |                        |     |                                                                 |             |     |                                                                 |              |     |                                                                         |            |     |                             |                        |     |                      |                        |     |                      |                      |     |                                    |                        |
| 010          | Fall transition on TCRCLK signal increments the TCR2 prescaler.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Falling edge                    |                                       |                                 |     |                                                                                                                                                                                                                                                                                                 |                        |     |                                                                 |             |     |                                                                 |              |     |                                                                         |            |     |                             |                        |     |                      |                        |     |                      |                      |     |                                    |                        |
| 011          | Rise or Fall transition on TCRCLK signal increments the TCR2 prescaler.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Both edges                      |                                       |                                 |     |                                                                                                                                                                                                                                                                                                 |                        |     |                                                                 |             |     |                                                                 |              |     |                                                                         |            |     |                             |                        |     |                      |                        |     |                      |                      |     |                                    |                        |
| 100          | DIV8 clock (eTPU clock / 8)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Do not use with AM > 0          |                                       |                                 |     |                                                                                                                                                                                                                                                                                                 |                        |     |                                                                 |             |     |                                                                 |              |     |                                                                         |            |     |                             |                        |     |                      |                        |     |                      |                      |     |                                    |                        |
| 101          | Do not use with AM=0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Do not use with AM > 0          |                                       |                                 |     |                                                                                                                                                                                                                                                                                                 |                        |     |                                                                 |             |     |                                                                 |              |     |                                                                         |            |     |                             |                        |     |                      |                        |     |                      |                      |     |                                    |                        |
| 110          | Do not use with AM=0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | No edge <sup>1</sup>            |                                       |                                 |     |                                                                                                                                                                                                                                                                                                 |                        |     |                                                                 |             |     |                                                                 |              |     |                                                                         |            |     |                             |                        |     |                      |                        |     |                      |                      |     |                                    |                        |
| 111          | TCR2 frozen, except as STAC client                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Do not use with AM > 0          |                                       |                                 |     |                                                                                                                                                                                                                                                                                                 |                        |     |                                                                 |             |     |                                                                 |              |     |                                                                         |            |     |                             |                        |     |                      |                        |     |                      |                      |     |                                    |                        |
| 3–4<br>TCRCF | <p>TCRCLK Signal Filter Control.</p> <p>This field controls the TCRCLK digital filter (see <a href="#">TCRCLK digital filter</a>), determining whether the TCRCLK signal input (after a synchronizer) is filtered with the same filter clock as the channel input signals (see <a href="#">Enhanced Digital Filter - EDF</a>) or uses the eTPU clock divided by 2, and also whether the TCRCLK digital filter works in integrator mode or two sample mode (see the following table).</p> <p><b>Table 35-23. TCRCLK Filter Clock/Mode</b></p> <table><tr><th>TCRCF</th><th>Filter Clock</th><th>Filter Mode</th></tr><tr><td>00</td><td>eTPU clock divided by 2</td><td>Two sample</td></tr><tr><td>01</td><td>Filter clock of the channels</td><td>Two sample</td></tr><tr><td>10</td><td>eTPU clock divided by 2</td><td>Integrator</td></tr><tr><td>11</td><td>Filter clock of the channels</td><td>Integrator</td></tr></table>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | TCRCF                           | Filter Clock                          | Filter Mode                     | 00  | eTPU clock divided by 2                                                                                                                                                                                                                                                                         | Two sample             | 01  | Filter clock of the channels                                    | Two sample  | 10  | eTPU clock divided by 2                                         | Integrator   | 11  | Filter clock of the channels                                            | Integrator |     |                             |                        |     |                      |                        |     |                      |                      |     |                                    |                        |
| TCRCF        | Filter Clock                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Filter Mode                     |                                       |                                 |     |                                                                                                                                                                                                                                                                                                 |                        |     |                                                                 |             |     |                                                                 |              |     |                                                                         |            |     |                             |                        |     |                      |                        |     |                      |                      |     |                                    |                        |
| 00           | eTPU clock divided by 2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Two sample                      |                                       |                                 |     |                                                                                                                                                                                                                                                                                                 |                        |     |                                                                 |             |     |                                                                 |              |     |                                                                         |            |     |                             |                        |     |                      |                        |     |                      |                      |     |                                    |                        |
| 01           | Filter clock of the channels                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Two sample                      |                                       |                                 |     |                                                                                                                                                                                                                                                                                                 |                        |     |                                                                 |             |     |                                                                 |              |     |                                                                         |            |     |                             |                        |     |                      |                        |     |                      |                      |     |                                    |                        |
| 10           | eTPU clock divided by 2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Integrator                      |                                       |                                 |     |                                                                                                                                                                                                                                                                                                 |                        |     |                                                                 |             |     |                                                                 |              |     |                                                                         |            |     |                             |                        |     |                      |                        |     |                      |                      |     |                                    |                        |
| 11           | Filter clock of the channels                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Integrator                      |                                       |                                 |     |                                                                                                                                                                                                                                                                                                 |                        |     |                                                                 |             |     |                                                                 |              |     |                                                                         |            |     |                             |                        |     |                      |                        |     |                      |                      |     |                                    |                        |
| 5–6<br>AM    | <p>Angle Mode Selection.</p> <p>This field enables the Enhanced Angle Counter logic to generate angle information (see <a href="#">EAC - eTPU Angle Counter</a>), and also select the tooth signal input and the channel used to process it, as shown in the following table. When EAC is not disabled by AM and neither TCR1 nor TCR2 are STAC Clients, the EAC</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                 |                                       |                                 |     |                                                                                                                                                                                                                                                                                                 |                        |     |                                                                 |             |     |                                                                 |              |     |                                                                         |            |     |                             |                        |     |                      |                        |     |                      |                      |     |                                    |                        |

*Table continues on the next page...*

**eTPU\_TBR\_TBCR\_ENG1 field descriptions (continued)**

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                 |                          |              |                          |    |                                   |                |  |    |             |              |   |    |                 |   |    |                 |   |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|--------------------------|--------------|--------------------------|----|-----------------------------------|----------------|--|----|-------------|--------------|---|----|-----------------|---|----|-----------------|---|
|                  | <p>(eTPU Angle Clock) hardware provides angle information to the channels using the TCR2 bus. When AM is reset (non-angle mode), the EAC operation is disabled, and its internal registers can be used as general purpose. For more information, see <a href="#">EAC - eTPU Angle Counter</a>.</p> <p><b>Table 35-22. AM - Angle Mode Selection</b></p> <table><tr><th>Value</th><th>TCR2 Value</th><th>Tooth signal</th><th>Tooth processing channel</th></tr><tr><td>00</td><td>Timebase (EAC operation disabled)</td><td colspan="2">Not applicable</td></tr><tr><td>01</td><td rowspan="3">Angle Ticks</td><td>TCRCLK input</td><td>0</td></tr><tr><td>10</td><td>Channel 1 input</td><td>1</td></tr><tr><td>11</td><td>Channel 2 input</td><td>2</td></tr></table> <p>If TCR1 or TCR2 is a STAC Bus Client (see <a href="#">STAC interface</a>), the EAC operation is not allowed, and AM must be set to 0.</p> <p><b>NOTE:</b> Changing AM may cause unexpected transition detections on the channel selected by AM, depending on the channel mode and state (see <a href="#">Transition detection and Time Base Capture</a>). If AM must be changed with GTBE=1, the recommended procedure is described in <a href="#">Restarting angle logic</a>.</p> | Value           | TCR2 Value               | Tooth signal | Tooth processing channel | 00 | Timebase (EAC operation disabled) | Not applicable |  | 01 | Angle Ticks | TCRCLK input | 0 | 10 | Channel 1 input | 1 | 11 | Channel 2 input | 2 |
| Value            | TCR2 Value                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Tooth signal    | Tooth processing channel |              |                          |    |                                   |                |  |    |             |              |   |    |                 |   |    |                 |   |
| 00               | Timebase (EAC operation disabled)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | Not applicable  |                          |              |                          |    |                                   |                |  |    |             |              |   |    |                 |   |    |                 |   |
| 01               | Angle Ticks                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | TCRCLK input    | 0                        |              |                          |    |                                   |                |  |    |             |              |   |    |                 |   |    |                 |   |
| 10               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Channel 1 input | 1                        |              |                          |    |                                   |                |  |    |             |              |   |    |                 |   |    |                 |   |
| 11               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Channel 2 input | 2                        |              |                          |    |                                   |                |  |    |             |              |   |    |                 |   |    |                 |   |
| 7<br>ATGC        | <p>Angle Tick Generator Clock.</p> <p>This bit selects the clock that provides the time interval unit for Angle Tick Generation. For details, see <a href="#">Generating the angle ticks</a>.</p> <p>0   TCR1 prescaler output provides time intervals for tick generation.</p> <p>1   TCR2 prescaler output provides time intervals for tick generation.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                 |                          |              |                          |    |                                   |                |  |    |             |              |   |    |                 |   |    |                 |   |
| 8–9<br>Reserved  | <p>Reserved.</p> <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                 |                          |              |                          |    |                                   |                |  |    |             |              |   |    |                 |   |    |                 |   |
| 10–15<br>TCR2P   | <p>Timer Count Register 2 Prescaler Control.</p> <p>These bits are part of the TCR2 clocking system (see <a href="#">Time bases</a>). TCR2 is clocked from the output of a prescaler. The prescaler divides its input by (TCR2P + 1) allowing frequency divisions from 1 to 64. The prescaler input is the eTPU clock divided by 8 (in gated or non-gated clock mode), or Internal Timebase input, or TCRCLK filtered input. This field has no effect on TCR2 in Angle Mode.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                 |                          |              |                          |    |                                   |                |  |    |             |              |   |    |                 |   |    |                 |   |
| 16–17<br>TCR1CTL | <p>TCR1 Clock/Gate Control.</p> <p>TCR1CTL is part of the TCR1 clocking system (see <a href="#">Time bases</a>). It determines, together with TCR1CS, the clock source for TCR1. TCR1 can count on detected rising edge of the TCRCLK signal, the eTPU clock, or the eTPU clock divided by 2 (see the table in the TCR1CS bit description). After reset TCRCLK signal is selected.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                 |                          |              |                          |    |                                   |                |  |    |             |              |   |    |                 |   |    |                 |   |
| 18<br>TCR1CS     | <p>TCR1 Clock Source</p> <p>TCR1CS provides the option to double the TCR1 incrementing speed, using eTPU clock as its clock source instead of eTPU clock / 2.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                 |                          |              |                          |    |                                   |                |  |    |             |              |   |    |                 |   |    |                 |   |

*Table continues on the next page...*

**eTPU\_TBR\_TBCR\_ENG1 field descriptions (continued)**

| Field                                                                                                                                     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |         |                                                                         |                             |    |   |                                                                     |    |   |          |    |   |          |    |   |                                                                         |    |   |                                                            |    |   |                                      |
|-------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|-------------------------------------------------------------------------|-----------------------------|----|---|---------------------------------------------------------------------|----|---|----------|----|---|----------|----|---|-------------------------------------------------------------------------|----|---|------------------------------------------------------------|----|---|--------------------------------------|
|                                                                                                                                           | <b>NOTE:</b> If ATGC=0, the clock source of the EAC angle tick generator will still be an even division of eTPU clock if TCR1CS=1, obeying to the field TCR1P as if TCR1CS=0 (see <a href="#">Angle Tick Generator</a> ).                                                                                                                                                                                                                                                                                                                                                                    |         |                                                                         |                             |    |   |                                                                     |    |   |          |    |   |          |    |   |                                                                         |    |   |                                                            |    |   |                                      |
|                                                                                                                                           | <b>Table 35-21. TCR1 Clock Source</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |         |                                                                         |                             |    |   |                                                                     |    |   |          |    |   |          |    |   |                                                                         |    |   |                                                            |    |   |                                      |
|                                                                                                                                           | <table><tr><th>TCR1CTL</th><th>TCR1CS</th><th>TCR1 Clock before prescaler</th></tr><tr><td>00</td><td>0</td><td>Selects TCRCLK as clock source for the TCR1 prescaler.<sup>1</sup></td></tr><tr><td>01</td><td>0</td><td>Reserved</td></tr><tr><td>01</td><td>1</td><td>Reserved</td></tr><tr><td>10</td><td>0</td><td>Selects eTPU clock divided by 2 as clock source for the TCR1 prescaler.</td></tr><tr><td>10</td><td>1</td><td>Selects eTPU clock as clock source for the TCR1 prescaler.</td></tr><tr><td>11</td><td>0</td><td>TCR1 frozen, except as a STAC client</td></tr></table> | TCR1CTL | TCR1CS                                                                  | TCR1 Clock before prescaler | 00 | 0 | Selects TCRCLK as clock source for the TCR1 prescaler. <sup>1</sup> | 01 | 0 | Reserved | 01 | 1 | Reserved | 10 | 0 | Selects eTPU clock divided by 2 as clock source for the TCR1 prescaler. | 10 | 1 | Selects eTPU clock as clock source for the TCR1 prescaler. | 11 | 0 | TCR1 frozen, except as a STAC client |
|                                                                                                                                           | TCR1CTL                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | TCR1CS  | TCR1 Clock before prescaler                                             |                             |    |   |                                                                     |    |   |          |    |   |          |    |   |                                                                         |    |   |                                                            |    |   |                                      |
|                                                                                                                                           | 00                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 0       | Selects TCRCLK as clock source for the TCR1 prescaler. <sup>1</sup>     |                             |    |   |                                                                     |    |   |          |    |   |          |    |   |                                                                         |    |   |                                                            |    |   |                                      |
|                                                                                                                                           | 01                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 0       | Reserved                                                                |                             |    |   |                                                                     |    |   |          |    |   |          |    |   |                                                                         |    |   |                                                            |    |   |                                      |
|                                                                                                                                           | 01                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 1       | Reserved                                                                |                             |    |   |                                                                     |    |   |          |    |   |          |    |   |                                                                         |    |   |                                                            |    |   |                                      |
|                                                                                                                                           | 10                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 0       | Selects eTPU clock divided by 2 as clock source for the TCR1 prescaler. |                             |    |   |                                                                     |    |   |          |    |   |          |    |   |                                                                         |    |   |                                                            |    |   |                                      |
|                                                                                                                                           | 10                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 1       | Selects eTPU clock as clock source for the TCR1 prescaler.              |                             |    |   |                                                                     |    |   |          |    |   |          |    |   |                                                                         |    |   |                                                            |    |   |                                      |
|                                                                                                                                           | 11                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 0       | TCR1 frozen, except as a STAC client                                    |                             |    |   |                                                                     |    |   |          |    |   |          |    |   |                                                                         |    |   |                                                            |    |   |                                      |
| 1. This selection must not be used in Angle Mode.                                                                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |         |                                                                         |                             |    |   |                                                                     |    |   |          |    |   |          |    |   |                                                                         |    |   |                                                            |    |   |                                      |
| 0 Use eTPU clock / 2 as TCR1 clock source before the prescaler, if that clock source is selected by TCR1CTL.                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |         |                                                                         |                             |    |   |                                                                     |    |   |          |    |   |          |    |   |                                                                         |    |   |                                                            |    |   |                                      |
| 1 Use eTPU clock as TCR1 clock source before the prescaler; can only be set in specific combinations with TCR1CTL, as shown in the table. |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |         |                                                                         |                             |    |   |                                                                     |    |   |          |    |   |          |    |   |                                                                         |    |   |                                                            |    |   |                                      |
| <b>NOTE:</b> TCR1CS=1 also makes the channel work on T2/T4 timing mode (see <a href="#">T2/T4 channel timing</a> ).                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |         |                                                                         |                             |    |   |                                                                     |    |   |          |    |   |          |    |   |                                                                         |    |   |                                                            |    |   |                                      |
| 19–23<br>Reserved                                                                                                                         | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |         |                                                                         |                             |    |   |                                                                     |    |   |          |    |   |          |    |   |                                                                         |    |   |                                                            |    |   |                                      |
| 24–31<br>TCR1P                                                                                                                            | Timer Count Register 1 Prescaler Control.<br><br>TCR1 is clocked from the output of a prescaler. The input to the prescaler is the internal eTPU clock divided by 2, eTPU clock, or the output of TCRCLK filter. The prescaler divides this input by (TCR1P + 1) allowing frequency divisions from 1 up to 256.                                                                                                                                                                                                                                                                              |         |                                                                         |                             |    |   |                                                                     |    |   |          |    |   |          |    |   |                                                                         |    |   |                                                            |    |   |                                      |

1. TCRCLK edges are not detected by the EAC logic, but they can still be detected by the channel 0 logic if AM=01.

2. This selection must not be used in Angle Mode.

### 35.8.2 eTPU Engine 1 Time Base 1 (TCR1) Visibility Register (eTPU\_TBR\_TB1R\_ENG1)

This register provides visibility of the TCR1 time base for host read access (see [Time bases](#)). This register is read-only. The value of the TCR1 time base shown can be driven by the TCR1 counter or imported from STAC bus, depending on the configuration set in [eTPU Engine 1 STAC Configuration Register \(eTPU\\_TBR\\_STAC\\_ENG1\)](#).

Address: 0h base + 24h offset = 24h

|       |   |   |   |   |   |   |   |   |      |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|------|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8    | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   | TCR1 |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |      |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0    | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**eTPU\_TBR\_TB1R\_ENG1 field descriptions**

| Field           | Description                                                                                              |
|-----------------|----------------------------------------------------------------------------------------------------------|
| 0–7<br>Reserved | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 8–31<br>TCR1    | TCR1 value.<br><br>TCR1 value used on matches and captures. See <a href="#">Time bases</a> .             |

### 35.8.3 eTPU Engine 1 Time Base 2 (TCR2) Visibility Register (eTPU\_TBR\_TB2R\_ENG1)

This register provides visibility of the TCR2 time base for host read access (see [Time bases](#)). This register is read-only. The value of the TCR2 time base shown can be driven by the TCR2 counter or imported from the STAC bus, depending on Angle Mode and STAC configurations set in [eTPU Engine 1 STAC Configuration Register \(eTPU\\_TBR\\_STAC\\_ENG1\)](#).

Address: 0h base + 28h offset = 28h

|       |   |   |   |   |   |   |   |   |      |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|------|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8    | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   | TCR2 |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |      |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0    | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |



**eTPU\_TBR\_TB2R\_ENG1 field descriptions**

| Field           | Description                                                                                              |
|-----------------|----------------------------------------------------------------------------------------------------------|
| 0–7<br>Reserved | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 8–31<br>TCR2    | TCR2 value.<br><br>TCR2 value used on matches and captures. See <a href="#">Time bases</a> .             |

**35.8.4 eTPU Engine 1 STAC Configuration Register (eTPU\_TBR\_STAC\_ENG1)**

This register configures the eTPU STAC bus operation as a STAC Server/Client module (see [STAC interface](#)).

Address: 0h base + 2Ch offset = 2Ch

|       |      |      |   |   |            |   |   |   |   |   |    |    |    |      |    |    |
|-------|------|------|---|---|------------|---|---|---|---|---|----|----|----|------|----|----|
| Bit   | 0    | 1    | 2 | 3 | 4          | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13   | 14 | 15 |
| R     | REN1 | RSC1 | 0 |   | SERVER_ID1 |   |   |   |   | 0 |    |    |    | SRV1 |    |    |
| W     |      |      |   |   |            |   |   |   |   |   |    |    |    |      |    |    |
| Reset | 0    | 0    | 0 | 0 | *          | * | * | * | 0 | 0 | 0  | 0  | 0  | 0    | 0  | 0  |

|       |      |      |    |    |            |    |    |    |    |    |    |    |    |      |    |    |
|-------|------|------|----|----|------------|----|----|----|----|----|----|----|----|------|----|----|
| Bit   | 16   | 17   | 18 | 19 | 20         | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29   | 30 | 31 |
| R     | REN2 | RSC2 | 0  |    | SERVER_ID2 |    |    |    |    | 0  |    |    |    | SRV2 |    |    |
| W     |      |      |    |    |            |    |    |    |    |    |    |    |    |      |    |    |
| Reset | 0    | 0    | 0  | 0  | *          | *  | *  | *  | 0  | 0  | 0  | 0  | 0  | 0    | 0  | 0  |

\* Notes:

- SERVER\_ID2 field: See the chip-specific eTPU information for the reset value.
- SERVER\_ID1 field: See the chip-specific eTPU information for the reset value.

**eTPU\_TBR\_STAC\_ENG1 field descriptions**

| Field     | Description                                                                                                                                                                                                                                                               |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>REN1 | TCR1 Resource Client/Server Operation Enable Bit.<br><br>This bit enables Client/Server operation to eTPU STAC resources. REN1 enables TCR1 STAC bus operations.<br><br>0 Server/Client Operation for TCR1 is disabled.<br>1 Server/Client Operation for TCR1 is enabled. |
| 1<br>RSC1 | TCR1 Resource Server/Client Assignment Bits.<br><br>This bit selects the eTPU data resource assignment to be used as Servers or Clients. For Server mode, MCU-dependent value determines the unique server address assigned to each TCR. See the chip-specific            |

*Table continues on the next page...*

**eTPU\_TBR\_STAC\_ENG1 field descriptions (continued)**

| Field               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                     | <p>eTPU information for the STAC server IDs. For a Client mode, the SRV1 field determines the Server address to which the Client listens.</p> <p><b>NOTE:</b> When TCR1 or TCR2 is configured as a STAC Bus Client (REN1/2 = 1, RSC1/2 = 0) the eTPU Angle Clock hardware cannot be used.</p> <p><b>NOTE:</b> RSC1 must not be changed when the REN1 bit is asserted.</p> <p>0 Resource Client operation.<br/>1 Resource Server operation.</p> |
| 2–3<br>Reserved     | <p>Reserved.</p> <p>This field is reserved.<br/>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                               |
| 4–7<br>SERVER_ID1   | <p>STAC ID 1.</p> <p>STAC Server ID (read-only MCU-dependent values; see the chip-specific eTPU information for the STAC server IDs) used for TCR1 when STAC servers are used.</p>                                                                                                                                                                                                                                                             |
| 8–11<br>Reserved    | <p>Reserved.</p> <p>This field is reserved.<br/>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                               |
| 12–15<br>SRV1       | <p>TCR1 Resource Server.</p> <p>This bit selects the address of the specific STAC Server to which the local TCR1 listens when configured as a STAC Client. SRV1 selects the STAC Server of TCR1.</p>                                                                                                                                                                                                                                           |
| 16<br>REN2          | <p>TCR2 Resource Client/Server Operation Enable Bit.</p> <p>This bit enables Client/Server operation to eTPU STAC resources. REN2 enables TCR2 STAC bus operations.</p> <p>0 Server/Client Operation for TCR2 is disabled.<br/>1 Server/Client Operation for TCR2 is enabled.</p>                                                                                                                                                              |
| 17<br>RSC2          | <p>TCR2 Resource Server/Client Assignment Bits.</p> <p><b>NOTE:</b> Either time or angle TCR2 values.</p> <p>0 Resource Client operation.<br/>1 Resource Server operation.</p>                                                                                                                                                                                                                                                                 |
| 18–19<br>Reserved   | <p>Reserved.</p> <p>This field is reserved.<br/>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                               |
| 20–23<br>SERVER_ID2 | <p>STAC ID 2.</p> <p>STAC Server ID (read-only MCU-dependent values; see the chip-specific eTPU information for the STAC server IDs) used for TCR2 when when STAC servers are used.</p>                                                                                                                                                                                                                                                        |
| 24–27<br>Reserved   | <p>Reserved.</p> <p>This field is reserved.<br/>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                               |

*Table continues on the next page...*

**eTPU\_TBR\_STAC\_ENG1 field descriptions (continued)**

| Field         | Description                                                                                                                                                                                            |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 28–31<br>SRV2 | TCR2 Resource Server.<br><br>This bit selects the address of the specific STAC Server to which the local TCR2 listens when configured as a STAC Client. SRV2 selects the Red Line STAC Server of TCR2. |

**35.8.5 eTPU Engine 2 Time Base Configuration Register (eTPU\_TBR\_TBCR\_ENG2)**

This register configures several timebase options.

Address: 0h base + 40h offset = 40h

|       |         |    |        |    |    |    |      |    |       |    |       |    |    |    |    |    |
|-------|---------|----|--------|----|----|----|------|----|-------|----|-------|----|----|----|----|----|
| Bit   | 0       | 1  | 2      | 3  | 4  | 5  | 6    | 7  | 8     | 9  | 10    | 11 | 12 | 13 | 14 | 15 |
| R     | TCR2CTL |    | TCRCF  |    | AM |    | ATGC |    | 0     |    | TCR2P |    |    |    |    |    |
| W     |         |    |        |    |    |    |      |    |       |    |       |    |    |    |    |    |
| Reset | 0       | 0  | 1      | 0  | 0  | 0  | 0    | 0  | 0     | 0  | 0     | 0  | 0  | 0  | 0  | 0  |
| Bit   | 16      | 17 | 18     | 19 | 20 | 21 | 22   | 23 | 24    | 25 | 26    | 27 | 28 | 29 | 30 | 31 |
| R     | TCR1CTL |    | TCR1CS | 0  |    |    |      |    | TCR1P |    |       |    |    |    |    |    |
| W     |         |    |        |    |    |    |      |    |       |    |       |    |    |    |    |    |
| Reset | 0       | 0  | 0      | 0  | 0  | 0  | 0    | 0  | 0     | 0  | 0     | 0  | 0  | 0  | 0  | 0  |

**eTPU\_TBR\_TBCR\_ENG2 field descriptions**

| Field          | Description                                                                                                                                                                                                                                                                                                                                                    |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–2<br>TCR2CTL | TCR2 Clock/Gate Control.<br><br>These bits are part of the TCR2 clocking system (see <a href="#">Time bases</a> ). They determine the clock source for TCR2 before the prescaler. TCR2 can count on any detected edge of the TCRCLK signal or use it for gating eTPU clock divided by 8. After reset - TCRCLK signal rising edge is selected. TCR2 can also be |

*Table continues on the next page...*

**eTPU\_TBR\_TBCR\_ENG2 field descriptions (continued)**

| Field        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                 |                                       |                                 |     |                                                                                                                                                                                                                                                                                                 |                        |     |                                                                 |             |     |                                                                 |              |     |                                                                         |            |     |                             |                        |     |                      |                        |     |                      |                      |     |                                    |                        |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------|---------------------------------------|---------------------------------|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|-----|-----------------------------------------------------------------|-------------|-----|-----------------------------------------------------------------|--------------|-----|-------------------------------------------------------------------------|------------|-----|-----------------------------|------------------------|-----|----------------------|------------------------|-----|----------------------|----------------------|-----|------------------------------------|------------------------|
|              | <p>clocked by the eTPU clock divided by 8. TCR2CTL also determines the TCRCLK edge selected for angle tooth detection in angle mode. See the following table.</p> <p><b>Table 35-32. TCR2 Clock Source</b></p> <table><tr><th>TCR2CTL</th><th>AM = 0<br/>TCR2 Clock before prescaler</th><th>AM &gt; 0<br/>Angle Tooth detection</th></tr><tr><td>000</td><td>Gated DIV8 clock (eTPU clock / 8). In this case, when the external TCRCLK signal is low, the DIV8 clock is blocked, preventing it from incrementing the TCR2 prescaler. When the external TCRCLK signal is high, TCR2 prescaler is incremented at the frequency of the eTPU clock divided by 8.</td><td>Do not use with AM &gt; 0</td></tr><tr><td>001</td><td>Rise transition on TCRCLK signal increments the TCR2 prescaler.</td><td>Rising edge</td></tr><tr><td>010</td><td>Fall transition on TCRCLK signal increments the TCR2 prescaler.</td><td>Falling edge</td></tr><tr><td>011</td><td>Rise or Fall transition on TCRCLK signal increments the TCR2 prescaler.</td><td>Both edges</td></tr><tr><td>100</td><td>DIV8 clock (eTPU clock / 8)</td><td>Do not use with AM &gt; 0</td></tr><tr><td>101</td><td>Do not use with AM=0</td><td>Do not use with AM &gt; 0</td></tr><tr><td>110</td><td>Do not use with AM=0</td><td>No edge<sup>1</sup></td></tr><tr><td>111</td><td>TCR2 frozen, except as STAC client</td><td>Do not use with AM &gt; 0</td></tr></table> <p>1. TCRCLK edges are not detected by the EAC logic, but they can still be detected by the channel 0 logic if AM=01.</p> | TCR2CTL                         | AM = 0<br>TCR2 Clock before prescaler | AM > 0<br>Angle Tooth detection | 000 | Gated DIV8 clock (eTPU clock / 8). In this case, when the external TCRCLK signal is low, the DIV8 clock is blocked, preventing it from incrementing the TCR2 prescaler. When the external TCRCLK signal is high, TCR2 prescaler is incremented at the frequency of the eTPU clock divided by 8. | Do not use with AM > 0 | 001 | Rise transition on TCRCLK signal increments the TCR2 prescaler. | Rising edge | 010 | Fall transition on TCRCLK signal increments the TCR2 prescaler. | Falling edge | 011 | Rise or Fall transition on TCRCLK signal increments the TCR2 prescaler. | Both edges | 100 | DIV8 clock (eTPU clock / 8) | Do not use with AM > 0 | 101 | Do not use with AM=0 | Do not use with AM > 0 | 110 | Do not use with AM=0 | No edge <sup>1</sup> | 111 | TCR2 frozen, except as STAC client | Do not use with AM > 0 |
| TCR2CTL      | AM = 0<br>TCR2 Clock before prescaler                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | AM > 0<br>Angle Tooth detection |                                       |                                 |     |                                                                                                                                                                                                                                                                                                 |                        |     |                                                                 |             |     |                                                                 |              |     |                                                                         |            |     |                             |                        |     |                      |                        |     |                      |                      |     |                                    |                        |
| 000          | Gated DIV8 clock (eTPU clock / 8). In this case, when the external TCRCLK signal is low, the DIV8 clock is blocked, preventing it from incrementing the TCR2 prescaler. When the external TCRCLK signal is high, TCR2 prescaler is incremented at the frequency of the eTPU clock divided by 8.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Do not use with AM > 0          |                                       |                                 |     |                                                                                                                                                                                                                                                                                                 |                        |     |                                                                 |             |     |                                                                 |              |     |                                                                         |            |     |                             |                        |     |                      |                        |     |                      |                      |     |                                    |                        |
| 001          | Rise transition on TCRCLK signal increments the TCR2 prescaler.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Rising edge                     |                                       |                                 |     |                                                                                                                                                                                                                                                                                                 |                        |     |                                                                 |             |     |                                                                 |              |     |                                                                         |            |     |                             |                        |     |                      |                        |     |                      |                      |     |                                    |                        |
| 010          | Fall transition on TCRCLK signal increments the TCR2 prescaler.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Falling edge                    |                                       |                                 |     |                                                                                                                                                                                                                                                                                                 |                        |     |                                                                 |             |     |                                                                 |              |     |                                                                         |            |     |                             |                        |     |                      |                        |     |                      |                      |     |                                    |                        |
| 011          | Rise or Fall transition on TCRCLK signal increments the TCR2 prescaler.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Both edges                      |                                       |                                 |     |                                                                                                                                                                                                                                                                                                 |                        |     |                                                                 |             |     |                                                                 |              |     |                                                                         |            |     |                             |                        |     |                      |                        |     |                      |                      |     |                                    |                        |
| 100          | DIV8 clock (eTPU clock / 8)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Do not use with AM > 0          |                                       |                                 |     |                                                                                                                                                                                                                                                                                                 |                        |     |                                                                 |             |     |                                                                 |              |     |                                                                         |            |     |                             |                        |     |                      |                        |     |                      |                      |     |                                    |                        |
| 101          | Do not use with AM=0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Do not use with AM > 0          |                                       |                                 |     |                                                                                                                                                                                                                                                                                                 |                        |     |                                                                 |             |     |                                                                 |              |     |                                                                         |            |     |                             |                        |     |                      |                        |     |                      |                      |     |                                    |                        |
| 110          | Do not use with AM=0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | No edge <sup>1</sup>            |                                       |                                 |     |                                                                                                                                                                                                                                                                                                 |                        |     |                                                                 |             |     |                                                                 |              |     |                                                                         |            |     |                             |                        |     |                      |                        |     |                      |                      |     |                                    |                        |
| 111          | TCR2 frozen, except as STAC client                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Do not use with AM > 0          |                                       |                                 |     |                                                                                                                                                                                                                                                                                                 |                        |     |                                                                 |             |     |                                                                 |              |     |                                                                         |            |     |                             |                        |     |                      |                        |     |                      |                      |     |                                    |                        |
| 3–4<br>TCRCF | <p>TCRCLK Signal Filter Control.</p> <p>This field controls the TCRCLK digital filter (see <a href="#">TCRCLK digital filter</a>, determining whether the TCRCLK signal input (after a synchronizer) is filtered with the same filter clock as the channel input signals (see <a href="#">Enhanced Digital Filter - EDF</a>) or uses the eTPU clock divided by 2, and also whether the TCRCLK digital filter works in integrator mode or two sample mode (see the following table).</p> <p><b>Table 35-31. TCRCLK Filter Clock/Mode</b></p> <table><tr><th>TCRCF</th><th>Filter Clock</th><th>Filter Mode</th></tr><tr><td>00</td><td>eTPU clock divided by 2</td><td>Two sample</td></tr><tr><td>01</td><td>Filter clock of the channels</td><td>Two sample</td></tr><tr><td>10</td><td>eTPU clock divided by 2</td><td>Integrator</td></tr><tr><td>11</td><td>Filter clock of the channels</td><td>Integrator</td></tr></table>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | TCRCF                           | Filter Clock                          | Filter Mode                     | 00  | eTPU clock divided by 2                                                                                                                                                                                                                                                                         | Two sample             | 01  | Filter clock of the channels                                    | Two sample  | 10  | eTPU clock divided by 2                                         | Integrator   | 11  | Filter clock of the channels                                            | Integrator |     |                             |                        |     |                      |                        |     |                      |                      |     |                                    |                        |
| TCRCF        | Filter Clock                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Filter Mode                     |                                       |                                 |     |                                                                                                                                                                                                                                                                                                 |                        |     |                                                                 |             |     |                                                                 |              |     |                                                                         |            |     |                             |                        |     |                      |                        |     |                      |                      |     |                                    |                        |
| 00           | eTPU clock divided by 2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Two sample                      |                                       |                                 |     |                                                                                                                                                                                                                                                                                                 |                        |     |                                                                 |             |     |                                                                 |              |     |                                                                         |            |     |                             |                        |     |                      |                        |     |                      |                      |     |                                    |                        |
| 01           | Filter clock of the channels                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Two sample                      |                                       |                                 |     |                                                                                                                                                                                                                                                                                                 |                        |     |                                                                 |             |     |                                                                 |              |     |                                                                         |            |     |                             |                        |     |                      |                        |     |                      |                      |     |                                    |                        |
| 10           | eTPU clock divided by 2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Integrator                      |                                       |                                 |     |                                                                                                                                                                                                                                                                                                 |                        |     |                                                                 |             |     |                                                                 |              |     |                                                                         |            |     |                             |                        |     |                      |                        |     |                      |                      |     |                                    |                        |
| 11           | Filter clock of the channels                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Integrator                      |                                       |                                 |     |                                                                                                                                                                                                                                                                                                 |                        |     |                                                                 |             |     |                                                                 |              |     |                                                                         |            |     |                             |                        |     |                      |                        |     |                      |                      |     |                                    |                        |
| 5–6<br>AM    | <p>Angle Mode Selection.</p> <p>This field enables the Enhanced Angle Counter logic to generate angle information (see <a href="#">EAC - eTPU Angle Counter</a>), and also select the tooth signal input and the channel used to process it, as shown in the following table. When EAC is not disabled by AM and neither TCR1 nor TCR2 are STAC Clients, the EAC (eTPU Angle Clock) hardware provides angle information to the channels using the TCR2 bus. When AM</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                 |                                       |                                 |     |                                                                                                                                                                                                                                                                                                 |                        |     |                                                                 |             |     |                                                                 |              |     |                                                                         |            |     |                             |                        |     |                      |                        |     |                      |                      |     |                                    |                        |

*Table continues on the next page...*

**eTPU\_TBR\_TBCR\_ENG2 field descriptions (continued)**

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                 |                          |              |                          |    |                                   |                |  |    |             |              |   |    |                 |   |    |                 |   |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|--------------------------|--------------|--------------------------|----|-----------------------------------|----------------|--|----|-------------|--------------|---|----|-----------------|---|----|-----------------|---|
|                  | <p>is reset (non-angle mode), the EAC operation is disabled, and its internal registers can be used as general purpose. For more information, see <a href="#">EAC - eTPU Angle Counter</a>.</p> <p><b>Table 35-30. AM - Angle Mode Selection</b></p> <table><tr><th>Value</th><th>TCR2 Value</th><th>Tooth signal</th><th>Tooth processing channel</th></tr><tr><td>00</td><td>Timebase (EAC operation disabled)</td><td colspan="2">Not applicable</td></tr><tr><td>01</td><td rowspan="3">Angle Ticks</td><td>TCRCLK input</td><td>0</td></tr><tr><td>10</td><td>Channel 1 input</td><td>1</td></tr><tr><td>11</td><td>Channel 2 input</td><td>2</td></tr></table> <p>If TCR1 or TCR2 is a STAC Bus Client (see <a href="#">STAC interface</a>), the EAC operation is not allowed, and AM must be set to 0.</p> <p><b>NOTE:</b> Changing AM may cause unexpected transition detections on the channel selected by AM, depending on the channel mode and state (see <a href="#">Transition detection and Time Base Capture</a>). If AM must be changed with GTBE=1, the recommended procedure is described in <a href="#">Restarting angle logic</a>.</p> | Value           | TCR2 Value               | Tooth signal | Tooth processing channel | 00 | Timebase (EAC operation disabled) | Not applicable |  | 01 | Angle Ticks | TCRCLK input | 0 | 10 | Channel 1 input | 1 | 11 | Channel 2 input | 2 |
| Value            | TCR2 Value                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Tooth signal    | Tooth processing channel |              |                          |    |                                   |                |  |    |             |              |   |    |                 |   |    |                 |   |
| 00               | Timebase (EAC operation disabled)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Not applicable  |                          |              |                          |    |                                   |                |  |    |             |              |   |    |                 |   |    |                 |   |
| 01               | Angle Ticks                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | TCRCLK input    | 0                        |              |                          |    |                                   |                |  |    |             |              |   |    |                 |   |    |                 |   |
| 10               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Channel 1 input | 1                        |              |                          |    |                                   |                |  |    |             |              |   |    |                 |   |    |                 |   |
| 11               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Channel 2 input | 2                        |              |                          |    |                                   |                |  |    |             |              |   |    |                 |   |    |                 |   |
| 7<br>ATGC        | <p>Angle Tick Generator Clock.</p> <p>This bit selects the clock that provides the time interval unit for Angle Tick Generation. For details, see <a href="#">Generating the angle ticks</a>.</p> <p>0 TCR1 prescaler output provides time intervals for tick generation.</p> <p>1 TCR2 prescaler output provides time intervals for tick generation.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                 |                          |              |                          |    |                                   |                |  |    |             |              |   |    |                 |   |    |                 |   |
| 8–9<br>Reserved  | <p>Reserved.</p> <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                 |                          |              |                          |    |                                   |                |  |    |             |              |   |    |                 |   |    |                 |   |
| 10–15<br>TCR2P   | <p>Timer Count Register 2 Prescaler Control.</p> <p>These bits are part of the TCR2 clocking system (see <a href="#">Time bases</a>). TCR2 is clocked from the output of a prescaler. The prescaler divides its input by (TCR2P + 1) allowing frequency divisions from 1 to 64. The prescaler input is the eTPU clock divided by 8 (in gated or non-gated clock mode), or Internal Timebase input, or TCRCLK filtered input. This field has no effect on TCR2 in Angle Mode.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                 |                          |              |                          |    |                                   |                |  |    |             |              |   |    |                 |   |    |                 |   |
| 16–17<br>TCR1CTL | <p>TCR1 Clock/Gate Control.</p> <p>TCR1CTL is part of the TCR1 clocking system (see <a href="#">Time bases</a>). It determines, together with TCR1CS, the clock source for TCR1. TCR1 can count on detected rising edge of the TCRCLK signal, the eTPU clock, or the eTPU clock divided by 2 (see the table in the TCR1CS bit description). After reset TCRCLK signal is selected.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                 |                          |              |                          |    |                                   |                |  |    |             |              |   |    |                 |   |    |                 |   |
| 18<br>TCR1CS     | <p>TCR1 Clock Source</p> <p>TCR1CS provides the option to double the TCR1 incrementing speed, using eTPU clock as its clock source instead of eTPU clock / 2.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                 |                          |              |                          |    |                                   |                |  |    |             |              |   |    |                 |   |    |                 |   |

*Table continues on the next page...*

**eTPU\_TBR\_TBCR\_ENG2 field descriptions (continued)**

| Field                                                                                                                                     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |         |                                                                         |                             |    |   |                                                                     |    |   |          |    |   |          |    |   |                                                                         |    |   |                                                            |    |   |                                      |
|-------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|-------------------------------------------------------------------------|-----------------------------|----|---|---------------------------------------------------------------------|----|---|----------|----|---|----------|----|---|-------------------------------------------------------------------------|----|---|------------------------------------------------------------|----|---|--------------------------------------|
|                                                                                                                                           | <b>NOTE:</b> If ATGC=0, the clock source of the EAC angle tick generator will still be an even division of eTPU clock if TCR1CS=1, obeying to the field TCR1P as if TCR1CS=0 (see <a href="#">Angle Tick Generator</a> ).                                                                                                                                                                                                                                                                                                                                                                    |         |                                                                         |                             |    |   |                                                                     |    |   |          |    |   |          |    |   |                                                                         |    |   |                                                            |    |   |                                      |
|                                                                                                                                           | <b>Table 35-29. TCR1 Clock Source</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |         |                                                                         |                             |    |   |                                                                     |    |   |          |    |   |          |    |   |                                                                         |    |   |                                                            |    |   |                                      |
|                                                                                                                                           | <table><tr><th>TCR1CTL</th><th>TCR1CS</th><th>TCR1 Clock before prescaler</th></tr><tr><td>00</td><td>0</td><td>Selects TCRCLK as clock source for the TCR1 prescaler.<sup>1</sup></td></tr><tr><td>01</td><td>0</td><td>Reserved</td></tr><tr><td>01</td><td>1</td><td>Reserved</td></tr><tr><td>10</td><td>0</td><td>Selects eTPU clock divided by 2 as clock source for the TCR1 prescaler.</td></tr><tr><td>10</td><td>1</td><td>Selects eTPU clock as clock source for the TCR1 prescaler.</td></tr><tr><td>11</td><td>0</td><td>TCR1 frozen, except as a STAC client</td></tr></table> | TCR1CTL | TCR1CS                                                                  | TCR1 Clock before prescaler | 00 | 0 | Selects TCRCLK as clock source for the TCR1 prescaler. <sup>1</sup> | 01 | 0 | Reserved | 01 | 1 | Reserved | 10 | 0 | Selects eTPU clock divided by 2 as clock source for the TCR1 prescaler. | 10 | 1 | Selects eTPU clock as clock source for the TCR1 prescaler. | 11 | 0 | TCR1 frozen, except as a STAC client |
|                                                                                                                                           | TCR1CTL                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | TCR1CS  | TCR1 Clock before prescaler                                             |                             |    |   |                                                                     |    |   |          |    |   |          |    |   |                                                                         |    |   |                                                            |    |   |                                      |
|                                                                                                                                           | 00                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 0       | Selects TCRCLK as clock source for the TCR1 prescaler. <sup>1</sup>     |                             |    |   |                                                                     |    |   |          |    |   |          |    |   |                                                                         |    |   |                                                            |    |   |                                      |
|                                                                                                                                           | 01                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 0       | Reserved                                                                |                             |    |   |                                                                     |    |   |          |    |   |          |    |   |                                                                         |    |   |                                                            |    |   |                                      |
|                                                                                                                                           | 01                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 1       | Reserved                                                                |                             |    |   |                                                                     |    |   |          |    |   |          |    |   |                                                                         |    |   |                                                            |    |   |                                      |
|                                                                                                                                           | 10                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 0       | Selects eTPU clock divided by 2 as clock source for the TCR1 prescaler. |                             |    |   |                                                                     |    |   |          |    |   |          |    |   |                                                                         |    |   |                                                            |    |   |                                      |
|                                                                                                                                           | 10                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 1       | Selects eTPU clock as clock source for the TCR1 prescaler.              |                             |    |   |                                                                     |    |   |          |    |   |          |    |   |                                                                         |    |   |                                                            |    |   |                                      |
|                                                                                                                                           | 11                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | 0       | TCR1 frozen, except as a STAC client                                    |                             |    |   |                                                                     |    |   |          |    |   |          |    |   |                                                                         |    |   |                                                            |    |   |                                      |
| 1. This selection must not be used in Angle Mode.                                                                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |         |                                                                         |                             |    |   |                                                                     |    |   |          |    |   |          |    |   |                                                                         |    |   |                                                            |    |   |                                      |
| 0 Use eTPU clock / 2 as TCR1 clock source before the prescaler, if that clock source is selected by TCR1CTL.                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |         |                                                                         |                             |    |   |                                                                     |    |   |          |    |   |          |    |   |                                                                         |    |   |                                                            |    |   |                                      |
| 1 Use eTPU clock as TCR1 clock source before the prescaler; can only be set in specific combinations with TCR1CTL, as shown in the table. |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |         |                                                                         |                             |    |   |                                                                     |    |   |          |    |   |          |    |   |                                                                         |    |   |                                                            |    |   |                                      |
| <b>NOTE:</b> TCR1CS=1 also makes the channel work on T2/T4 timing mode (see <a href="#">T2/T4 channel timing</a> ).                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |         |                                                                         |                             |    |   |                                                                     |    |   |          |    |   |          |    |   |                                                                         |    |   |                                                            |    |   |                                      |
| 19–23<br>Reserved                                                                                                                         | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |         |                                                                         |                             |    |   |                                                                     |    |   |          |    |   |          |    |   |                                                                         |    |   |                                                            |    |   |                                      |
| 24–31<br>TCR1P                                                                                                                            | Timer Count Register 1 Prescaler Control.<br><br>TCR1 is clocked from the output of a prescaler. The input to the prescaler is the internal eTPU clock divided by 2, eTPU clock, or the output of TCRCLK filter. The prescaler divides this input by (TCR1P + 1) allowing frequency divisions from 1 up to 256.                                                                                                                                                                                                                                                                              |         |                                                                         |                             |    |   |                                                                     |    |   |          |    |   |          |    |   |                                                                         |    |   |                                                            |    |   |                                      |

1. TCRCLK edges are not detected by the EAC logic, but they can still be detected by the channel 0 logic if AM=01.

2. This selection must not be used in Angle Mode.

### 35.8.6 eTPU Engine 2 Time Base 1 (TCR1) Visibility Register (eTPU\_TBR\_TB1R\_ENG2)

This register provides visibility of the TCR1 time base for host read access (see [Time bases](#)). This register is read-only. The value of the TCR1 time base shown can be driven by the TCR1 counter or imported from STAC bus, depending on the configuration set in [eTPU Engine 2 STAC Configuration Register \(eTPU\\_TBR\\_STAC\\_ENG2\)](#).

Address: 0h base + 44h offset = 44h

|       |   |   |   |   |   |   |   |   |      |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|------|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8    | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   | TCR1 |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |      |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0    | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**eTPU\_TBR\_TB1R\_ENG2 field descriptions**

| Field           | Description                                                                                              |
|-----------------|----------------------------------------------------------------------------------------------------------|
| 0–7<br>Reserved | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 8–31<br>TCR1    | TCR1 value.<br><br>TCR1 value used on matches and captures. See <a href="#">Time bases</a> .             |

### 35.8.7 eTPU Engine 2 Time Base 2 (TCR2) Visibility Register (eTPU\_TBR\_TB2R\_ENG2)

This register provides visibility of the TCR2 time base for host read access (see [Time bases](#)). This register is read-only. The value of the TCR2 time base shown can be driven by the TCR2 counter or imported from the STAC bus, depending on Angle Mode and STAC configurations set in [eTPU Engine 2 STAC Configuration Register \(eTPU\\_TBR\\_STAC\\_ENG2\)](#).

Address: 0h base + 48h offset = 48h

|       |   |   |   |   |   |   |   |   |      |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|------|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8    | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   | TCR2 |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |      |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0    | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**eTPU\_TBR\_TB2R\_ENG2 field descriptions**

| Field           | Description                                                                                              |
|-----------------|----------------------------------------------------------------------------------------------------------|
| 0–7<br>Reserved | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 8–31<br>TCR2    | TCR2 value.<br><br>TCR2 value used on matches and captures. See <a href="#">Time bases</a> .             |

**35.8.8 eTPU Engine 2 STAC Configuration Register (eTPU\_TBR\_STAC\_ENG2)**

This register configures the eTPU STAC bus operation as a STAC Server/Client module (see [STAC interface](#)).

Address: 0h base + 4Ch offset = 4Ch

|       |      |      |   |   |            |   |   |   |   |   |    |    |      |    |    |    |
|-------|------|------|---|---|------------|---|---|---|---|---|----|----|------|----|----|----|
| Bit   | 0    | 1    | 2 | 3 | 4          | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12   | 13 | 14 | 15 |
| R     |      |      | 0 |   | SERVER_ID1 |   |   |   | 0 |   |    |    | SRV1 |    |    |    |
| W     | REN1 | RSC1 |   |   |            |   |   |   |   |   |    |    |      |    |    |    |
| Reset | 0    | 0    | 0 | 0 | *          | * | * | * | 0 | 0 | 0  | 0  | 0    | 0  | 0  | 0  |

|       |      |      |    |    |            |    |    |    |    |    |    |    |      |    |    |    |
|-------|------|------|----|----|------------|----|----|----|----|----|----|----|------|----|----|----|
| Bit   | 16   | 17   | 18 | 19 | 20         | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28   | 29 | 30 | 31 |
| R     |      |      | 0  |    | SERVER_ID2 |    |    |    | 0  |    |    |    | SRV2 |    |    |    |
| W     | REN2 | RSC2 |    |    |            |    |    |    |    |    |    |    |      |    |    |    |
| Reset | 0    | 0    | 0  | 0  | *          | *  | *  | *  | 0  | 0  | 0  | 0  | 0    | 0  | 0  | 0  |

\* Notes:

- SERVER\_ID2 field: See the chip-specific eTPU information for the reset value.
- SERVER\_ID1 field: See the chip-specific eTPU information for the reset value.

**eTPU\_TBR\_STAC\_ENG2 field descriptions**

| Field     | Description                                                                                                                                                                                                                                                               |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>REN1 | TCR1 Resource Client/Server Operation Enable Bit.<br><br>This bit enables Client/Server operation to eTPU STAC resources. REN1 enables TCR1 STAC bus operations.<br><br>0 Server/Client Operation for TCR1 is disabled.<br>1 Server/Client Operation for TCR1 is enabled. |
| 1<br>RSC1 | TCR1 Resource Server/Client Assignment Bits.<br><br>This bit selects the eTPU data resource assignment to be used as Servers or Clients. For Server mode, MCU-dependent value determines the unique server address assigned to each TCR. See the chip-specific            |

*Table continues on the next page...*



**eTPU\_TBR\_STAC\_ENG2 field descriptions (continued)**

| Field               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                     | <p>eTPU information for the STAC server IDs. For a Client mode, the SRV1 field determines the Server address to which the Client listens.</p> <p><b>NOTE:</b> Resource identifies any parameter that changes along the time and can be exported / imported from other device. In eTPU context, a resource can be TCR1 or TCR2 (either Time or Angle values).</p> <p><b>NOTE:</b> When TCR1 or TCR2 is configured as a STAC Bus Client (REN2 = 1, RSC2 = 0) the eTPU Angle Clock hardware cannot be used.</p> <p><b>NOTE:</b> RSC1 must not be changed when the REN1 bit is asserted.</p> <p>0 TCR1 Client operation.<br/>1 TCR1 Server operation.</p> |
| 2–3<br>Reserved     | <p>Reserved.</p> <p>This field is reserved.<br/>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 4–7<br>SERVER_ID1   | <p>STAC ID 1.</p> <p>STAC Server ID (read-only MCU-dependent values; see the chip-specific eTPU information for the STAC server IDs) used for TCR1 when STAC servers are used.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 8–11<br>Reserved    | <p>Reserved.</p> <p>This field is reserved.<br/>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 12–15<br>SRV1       | <p>TCR1 Resource Server.</p> <p>This bit selects the address of the specific STAC Server to which the local TCR1 listens when configured as a STAC Client. SRV1 selects the STAC Server of TCR1.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 16<br>REN2          | <p>TCR2 Resource Client/Server Operation Enable Bit.</p> <p>This bit enables Client/Server operation to eTPU STAC resources. REN2 enables TCR2 STAC bus operations.</p> <p>0 Server/Client Operation for resource 2 is disabled.<br/>1 Server/Client Operation for resource 2 is enabled.</p>                                                                                                                                                                                                                                                                                                                                                         |
| 17<br>RSC2          | <p>TCR2 Resource Server/Client Assignment Bits.</p> <p><b>NOTE:</b> Resource identifies any parameter that changes along the time and can be exported / imported from other device. In eTPU context, a resource can be TCR1 or TCR2 (either Time or Angle values).</p> <p>0 Resource Client operation.<br/>1 Resource Server operation.</p>                                                                                                                                                                                                                                                                                                           |
| 18–19<br>Reserved   | <p>Reserved.</p> <p>This field is reserved.<br/>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 20–23<br>SERVER_ID2 | <p>STAC ID 2.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

*Table continues on the next page...*

**eTPU\_TBR\_STAC\_ENG2 field descriptions (continued)**

| Field             | Description                                                                                                                                                                                            |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | STAC Server ID (read-only MCU-dependent values; see the chip-specific eTPU information for the STAC server IDs) used for TCR2 when STAC servers are used.                                              |
| 24–27<br>Reserved | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                               |
| 28–31<br>SRV2     | TCR2 Resource Server.<br><br>This bit selects the address of the specific STAC Server to which the local TCR2 listens when configured as a STAC Client. SRV2 selects the Red Line STAC Server of TCR2. |

## 35.9 Engine memory map and registers

This section gathers registers that are engine-related, other than [eTPU Engine 1 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG1\)](#) and [eTPU Engine 2 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG2\)](#).

**eTPU\_ER memory map**

| Address offset (hex) | Register name                                                 | Width (in bits) | Access | Reset value | Section/ page               |
|----------------------|---------------------------------------------------------------|-----------------|--------|-------------|-----------------------------|
| 60                   | eTPU Watchdog Timer Register for Engine 1 (eTPU_ER_WDTR_ENG1) | 32              | R/W    | 0000_0000h  | <a href="#">35.9.1/1334</a> |
| 68                   | eTPU Idle Register for Engine 1 (eTPU_ER_IDLER_ENG1)          | 32              | R/W    | Undefined   | <a href="#">35.9.2/1336</a> |
| 70                   | eTPU Watchdog Timer Register for Engine 2 (eTPU_ER_WDTR_ENG2) | 32              | R/W    | 0000_0000h  | <a href="#">35.9.3/1337</a> |
| 78                   | eTPU Idle Register for Engine 2 (eTPU_ER_IDLER_ENG2)          | 32              | R/W    | Undefined   | <a href="#">35.9.4/1338</a> |

### 35.9.1 eTPU Watchdog Timer Register for Engine 1 (eTPU\_ER\_WDTR\_ENG1)

This register configures the watchdog timer for the engine.

Address: 0h base + 60h offset = 60h

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
| R     |    |    | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

## eTPU\_ER\_WDTR\_ENG1 field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–1<br>WDM       | <p>Watchdog Mode.</p> <p>WDM selects the Watchdog operation mode. For more information on the Watchdog operation, see <a href="#">Watchdog</a>.</p> <p><b>NOTE:</b> Before a new mode is configured, all conditions below must apply:</p> <ol style="list-style-type: none"> <li>1. All channels must be disabled (<a href="#">CPR</a> = 00).</li> <li>2. No thread must be executing (<a href="#">eTPU Channel Service Status Register for Engine 1</a> (<a href="#">eTPU_GCR_CSSR_ENG1</a>) reads 0).</li> <li>3. The watchdog must be disabled (WDM=00).</li> </ol> <p>00 Disabled<br/>01 Reserved<br/>10 Thread length<br/>11 Busy length</p>   |
| 2<br>Reserved    | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 3<br>WMO         | <p>Watchdog Monitor Operation.</p> <p>This bit enables the monitor operation of the watchdog, inhibiting the thread ending and channel disabling in case of a timeout. The Global Exception occurs on a watchdog timeout, independently of WMO.</p> <p><b>NOTE:</b> <a href="#">WDSn</a>, <a href="#">WDTO1</a>, <a href="#">WDTO2</a>, and the Global Exception always asserts in case of a watchdog timeout, independently of WMO.</p> <p>0 Force the end of the running thread and disable the service channel in case of a timeout.<br/>1 Do not force the end of the running thread, nor disable the service channel in case of a timeout.</p> |
| 4–15<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 16–31<br>WDCNT   | <p>Watchdog Count.</p> <p>This field indicates the maximum number of microcycles allowed for a thread (in thread length mode) or a sequence of threads (in busy length mode) before the current running thread is forced to end. For more information on Watchdog operation, see <a href="#">Watchdog</a>.</p> <p><b>NOTE:</b> The TST microcycles are also counted by the watchdog.</p> <p><b>NOTE:</b> Fields WDCNT and WMO must not be changed when the watchdog is enabled.</p>                                                                                                                                                                 |

### 35.9.2 eTPU Idle Register for Engine 1 (eTPU\_ER\_IDLER\_ENG1)

This register counts the microcycles in which the microengine is idle (see [Idle Counter](#)

Address: 0h base + 68h offset = 68h

|       |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0        | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | IDLE_CNT |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*       | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |

|       |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |
|-------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|
| Bit   | 16       | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31   |
| R     | IDLE_CNT |    |    |    |    |    |    |    |    |    |    |    |    |    |    | ICLR |
| W     |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |
| Reset | x*       | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x*   |

- \* Notes:
- x = Undefined at reset.

#### eTPU\_ER\_IDLER\_ENG1 field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                             |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–30<br>IDLE_CNT | <p>Idle Count.</p> <p>This is a freerunning count of the number of idle microcycles in the microengine. For more information on idle counter operation, see <a href="#">Idle Counter</a> .</p> <p><b>NOTE:</b> When the register is read, the ICLR bit is concatenated with the IDLE_CNT field (as the LSB) to form a 32-bit value.</p> |
| 31<br>ICLR       | <p>Idle Clear.</p> <p>Writing a 1 to this bit clears the IDLE_CNT field.</p> <p><b>NOTE:</b> When the register is read, the ICLR bit is concatenated with the IDLE_CNT field (as the LSB) to form a 32-bit value.</p> <p>0 Do not clear the IDLE_CNT field.</p> <p>1 Clear the IDLE_CNT field.</p>                                      |

### 35.9.3 eTPU Watchdog Timer Register for Engine 2 (eTPU\_ER\_WDTR\_ENG2)

This register configures the watchdog timer for the engine.

Address: 0h base + 70h offset = 70h

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
| R     |    |    | 0  |    |    |    |    |    |    |    | 0  |    |    |    |    |    |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

#### eTPU\_ER\_WDTR\_ENG2 field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–1<br>WDM       | <p>Watchdog Mode.</p> <p>WDM selects the Watchdog operation mode. For more information on the Watchdog operation, see <a href="#">Watchdog</a>.</p> <p><b>NOTE:</b> Before a new mode is configured, all conditions below must apply:</p> <ol style="list-style-type: none"> <li>1. All channels must be disabled (<a href="#">CPR</a> = 00).</li> <li>2. No thread must be executing (<a href="#">eTPU Channel Service Status Register for Engine 2 (eTPU_GCR_CSSR_ENG2)</a> reads 0).</li> <li>3. The watchdog must be disabled (WDM=00).</li> </ol> <p>00 Disabled<br/>01 Reserved<br/>10 Thread length<br/>11 Busy length</p>                  |
| 2<br>Reserved    | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 3<br>WMO         | <p>Watchdog Monitor Operation.</p> <p>This bit enables the monitor operation of the watchdog, inhibiting the thread ending and channel disabling in case of a timeout. The Global Exception occurs on a watchdog timeout, independently of WMO.</p> <p><b>NOTE:</b> <a href="#">WDSn</a>, <a href="#">WDTO1</a>, <a href="#">WDTO2</a>, and the Global Exception always assert in case of a watchdog timeout, independently of WMO.</p> <p>0 Force the end of the running thread and disable the service channel in case of a timeout.<br/>1 Do not force the end of the running thread, nor disable the service channel in case of a timeout.</p> |
| 4–15<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 16–31<br>WDCNT   | Watchdog Count.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

Table continues on the next page...

**eTPU\_ER\_WDTR\_ENG2 field descriptions (continued)**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | <p>This field indicates the maximum number of microcycles allowed for a thread (in thread length mode) or a sequence of threads (in busy length mode) before the current running thread is forced to end. For more information on Watchdog operation, see <a href="#">Watchdog</a>.</p> <p><b>NOTE:</b> The TST microcycles are also counted by the watchdog.</p> <p><b>NOTE:</b> Fields WDCNT and WMO must not be changed when the watchdog is enabled.</p> |

**35.9.4 eTPU Idle Register for Engine 2 (eTPU\_ER\_IDLER\_ENG2)**

This register counts the microcycles in which the microengine is idle (see [Idle Counter](#) ).

Address: 0h base + 78h offset = 78h

|       |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |      |
|-------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--|--|--|--|--|--|--|--|--|--|--|--|--|--|------|
| Bit   | 0        | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |      |
| R     | IDLE_CNT |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |      |
| W     |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |      |
| Reset | x*       | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |      |
| Bit   | 16       | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |      |
| R     | IDLE_CNT |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ICLR |
| W     |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |      |
| Reset | x*       | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |      |

\* Notes:

- x = Undefined at reset.

**eTPU\_ER\_IDLER\_ENG2 field descriptions**

| Field            | Description                                                                                                                                                                                                                                                                                                                             |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–30<br>IDLE_CNT | <p>Idle Count.</p> <p>This is a freerunning count of the number of idle microcycles in the microengine. For more information on idle counter operation, see <a href="#">Idle Counter</a> .</p> <p><b>NOTE:</b> When the register is read, the ICLR bit is concatenated with the IDLE_CNT field (as the LSB) to form a 32-bit value.</p> |
| 31<br>ICLR       | <p>Idle Clear.</p> <p>Writing a 1 to this bit clears the IDLE_CNT field.</p> <p><b>NOTE:</b> When the register is read, the ICLR bit is concatenated with the IDLE_CNT field (as the LSB) to form a 32-bit value.</p> <p>0 Do not clear the IDLE_CNT field.</p> <p>1 Clear the IDLE_CNT field.</p>                                      |

## 35.10 Latency Monitor memory map and registers

**eTPU\_LMR memory map**

| Address offset (hex) | Register name                                                                  | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|--------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 80                   | eTPU Latency Monitor Configuration Register for Engine 1 (eTPU_LMR_LM0CR_ENG1) | 32              | R/W    | 0000_0000h  | <a href="#">35.10.1/1340</a> |
| 84                   | eTPU Latency Monitor Measure Register for Engine 1 (eTPU_LMR_LM0MR_ENG1)       | 32              | R/W    | 0000_0000h  | <a href="#">35.10.2/1342</a> |
| 88                   | eTPU Latency Monitor Configuration Register for Engine 1 (eTPU_LMR_LM1CR_ENG1) | 32              | R/W    | 0000_0000h  | <a href="#">35.10.1/1340</a> |
| 8C                   | eTPU Latency Monitor Measure Register for Engine 1 (eTPU_LMR_LM1MR_ENG1)       | 32              | R/W    | 0000_0000h  | <a href="#">35.10.2/1342</a> |
| 90                   | eTPU Latency Monitor Configuration Register for Engine 1 (eTPU_LMR_LM2CR_ENG1) | 32              | R/W    | 0000_0000h  | <a href="#">35.10.1/1340</a> |
| 94                   | eTPU Latency Monitor Measure Register for Engine 1 (eTPU_LMR_LM2MR_ENG1)       | 32              | R/W    | 0000_0000h  | <a href="#">35.10.2/1342</a> |
| 98                   | eTPU Latency Monitor Configuration Register for Engine 1 (eTPU_LMR_LM3CR_ENG1) | 32              | R/W    | 0000_0000h  | <a href="#">35.10.1/1340</a> |
| 9C                   | eTPU Latency Monitor Measure Register for Engine 1 (eTPU_LMR_LM3MR_ENG1)       | 32              | R/W    | 0000_0000h  | <a href="#">35.10.2/1342</a> |
| A0                   | eTPU Latency Monitor Configuration Register for Engine 1 (eTPU_LMR_LM4CR_ENG1) | 32              | R/W    | 0000_0000h  | <a href="#">35.10.1/1340</a> |
| A4                   | eTPU Latency Monitor Measure Register for Engine 1 (eTPU_LMR_LM4MR_ENG1)       | 32              | R/W    | 0000_0000h  | <a href="#">35.10.2/1342</a> |
| A8                   | eTPU Latency Monitor Configuration Register for Engine 1 (eTPU_LMR_LM5CR_ENG1) | 32              | R/W    | 0000_0000h  | <a href="#">35.10.1/1340</a> |
| AC                   | eTPU Latency Monitor Measure Register for Engine 1 (eTPU_LMR_LM5MR_ENG1)       | 32              | R/W    | 0000_0000h  | <a href="#">35.10.2/1342</a> |
| C0                   | eTPU Latency Monitor Configuration Register for Engine 2 (eTPU_LMR_LM0CR_ENG2) | 32              | R/W    | 0000_0000h  | <a href="#">35.10.3/1343</a> |
| C4                   | eTPU Latency Monitor Measure Register for Engine 2 (eTPU_LMR_LM0MR_ENG2)       | 32              | R/W    | 0000_0000h  | <a href="#">35.10.4/1344</a> |
| C8                   | eTPU Latency Monitor Configuration Register for Engine 2 (eTPU_LMR_LM1CR_ENG2) | 32              | R/W    | 0000_0000h  | <a href="#">35.10.3/1343</a> |
| CC                   | eTPU Latency Monitor Measure Register for Engine 2 (eTPU_LMR_LM1MR_ENG2)       | 32              | R/W    | 0000_0000h  | <a href="#">35.10.4/1344</a> |
| D0                   | eTPU Latency Monitor Configuration Register for Engine 2 (eTPU_LMR_LM2CR_ENG2) | 32              | R/W    | 0000_0000h  | <a href="#">35.10.3/1343</a> |
| D4                   | eTPU Latency Monitor Measure Register for Engine 2 (eTPU_LMR_LM2MR_ENG2)       | 32              | R/W    | 0000_0000h  | <a href="#">35.10.4/1344</a> |
| D8                   | eTPU Latency Monitor Configuration Register for Engine 2 (eTPU_LMR_LM3CR_ENG2) | 32              | R/W    | 0000_0000h  | <a href="#">35.10.3/1343</a> |
| DC                   | eTPU Latency Monitor Measure Register for Engine 2 (eTPU_LMR_LM3MR_ENG2)       | 32              | R/W    | 0000_0000h  | <a href="#">35.10.4/1344</a> |

Table continues on the next page...

### eTPU\_LMR memory map (continued)

| Address offset (hex) | Register name                                                                  | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|--------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| E0                   | eTPU Latency Monitor Configuration Register for Engine 2 (eTPU_LMR_LM4CR_ENG2) | 32              | R/W    | 0000_0000h  | <a href="#">35.10.3/1343</a> |
| E4                   | eTPU Latency Monitor Measure Register for Engine 2 (eTPU_LMR_LM4MR_ENG2)       | 32              | R/W    | 0000_0000h  | <a href="#">35.10.4/1344</a> |
| E8                   | eTPU Latency Monitor Configuration Register for Engine 2 (eTPU_LMR_LM5CR_ENG2) | 32              | R/W    | 0000_0000h  | <a href="#">35.10.3/1343</a> |
| EC                   | eTPU Latency Monitor Measure Register for Engine 2 (eTPU_LMR_LM5MR_ENG2)       | 32              | R/W    | 0000_0000h  | <a href="#">35.10.4/1344</a> |

## 35.10.1 eTPU Latency Monitor Configuration Register for Engine 1 (eTPU\_LMR\_LMnCR\_ENG1)

This register configures the operation of individual latency monitors.

Address: 0h base + 80h offset + (8d × i), where i=0d to 5d

| Bit   | 0   | 1 | 2 | 3 | 4 | 5     | 6 | 7 | 8 | 9 | 10 | 11      | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|-------|-----|---|---|---|---|-------|---|---|---|---|----|---------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| R     | LMM |   | 0 |   |   | SRSEL |   |   | 0 |   |    | MONCHAN |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |     |   |   |   |   |       |   |   |   |   |    |         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0   | 0 | 0 | 0 | 0 | 0     | 0 | 0 | 0 | 0 | 0  | 0       | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

### eTPU\_LMR\_LMnCR\_ENG1 field descriptions

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–1<br>LMM      | <p>Latency Monitor Mode.</p> <p>This field selects the operation mode of the latency monitor. See the Latency Monitor section for detailed mode descriptions.</p> <p><b>NOTE:</b> LMM must be in disabled mode to be changed to channel or priority mode.</p> <p><b>CPR</b> must not be changed while LMM of any Latency Monitor is in priority mode.</p> <p>00 Disabled mode<br/>01 Channel mode<br/>10 Priority mode<br/>11 Reserved</p> |
| 2–4<br>Reserved | <p>Reserved.</p> <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                        |
| 5–7<br>SRSEL    | Service Request Selection                                                                                                                                                                                                                                                                                                                                                                                                                  |

Table continues on the next page...



**eTPU\_LMR\_LMnCR\_ENG1 field descriptions (continued)**

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                           |              |               |     |                         |                        |     |                            |                           |     |                     |                         |     |                  |          |     |                                |     |                             |     |                                |     |                   |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|--------------|---------------|-----|-------------------------|------------------------|-----|----------------------------|---------------------------|-----|---------------------|-------------------------|-----|------------------|----------|-----|--------------------------------|-----|-----------------------------|-----|--------------------------------|-----|-------------------|
|                  | <p>This field selects which service requests are monitored. The SRSEL value has different meanings in channel and priority modes, according to the following table.</p> <p><b>Table 35-45. SRSEL - Service Request Selection</b></p> <table><tr><th>SRSEL</th><th>Channel Mode</th><th>Priority Mode</th></tr><tr><td>001</td><td>Link SRs monitored only</td><td>Low priority monitored</td></tr><tr><td>010</td><td>Channel SRs monitored only</td><td>Medium priority monitored</td></tr><tr><td>100</td><td>HSRs monitored only</td><td>High priority monitored</td></tr><tr><td>000</td><td>No SRs monitored</td><td rowspan="5">Reserved</td></tr><tr><td>011</td><td>Channel and Link SRs monitored</td></tr><tr><td>101</td><td>HSRs and Link SRs monitored</td></tr><tr><td>110</td><td>HSRs and Channel SRs monitored</td></tr><tr><td>111</td><td>All SRs monitored</td></tr></table> | SRSEL                     | Channel Mode | Priority Mode | 001 | Link SRs monitored only | Low priority monitored | 010 | Channel SRs monitored only | Medium priority monitored | 100 | HSRs monitored only | High priority monitored | 000 | No SRs monitored | Reserved | 011 | Channel and Link SRs monitored | 101 | HSRs and Link SRs monitored | 110 | HSRs and Channel SRs monitored | 111 | All SRs monitored |
| SRSEL            | Channel Mode                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Priority Mode             |              |               |     |                         |                        |     |                            |                           |     |                     |                         |     |                  |          |     |                                |     |                             |     |                                |     |                   |
| 001              | Link SRs monitored only                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Low priority monitored    |              |               |     |                         |                        |     |                            |                           |     |                     |                         |     |                  |          |     |                                |     |                             |     |                                |     |                   |
| 010              | Channel SRs monitored only                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Medium priority monitored |              |               |     |                         |                        |     |                            |                           |     |                     |                         |     |                  |          |     |                                |     |                             |     |                                |     |                   |
| 100              | HSRs monitored only                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | High priority monitored   |              |               |     |                         |                        |     |                            |                           |     |                     |                         |     |                  |          |     |                                |     |                             |     |                                |     |                   |
| 000              | No SRs monitored                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Reserved                  |              |               |     |                         |                        |     |                            |                           |     |                     |                         |     |                  |          |     |                                |     |                             |     |                                |     |                   |
| 011              | Channel and Link SRs monitored                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                           |              |               |     |                         |                        |     |                            |                           |     |                     |                         |     |                  |          |     |                                |     |                             |     |                                |     |                   |
| 101              | HSRs and Link SRs monitored                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                           |              |               |     |                         |                        |     |                            |                           |     |                     |                         |     |                  |          |     |                                |     |                             |     |                                |     |                   |
| 110              | HSRs and Channel SRs monitored                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                           |              |               |     |                         |                        |     |                            |                           |     |                     |                         |     |                  |          |     |                                |     |                             |     |                                |     |                   |
| 111              | All SRs monitored                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                           |              |               |     |                         |                        |     |                            |                           |     |                     |                         |     |                  |          |     |                                |     |                             |     |                                |     |                   |
| 8–10<br>Reserved | <p>Reserved.</p> <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                           |              |               |     |                         |                        |     |                            |                           |     |                     |                         |     |                  |          |     |                                |     |                             |     |                                |     |                   |
| 11–15<br>MONCHAN | <p>Monitored Channel.</p> <p>When the latency monitor is configured for channel mode, this field selects which channel has the latency monitored. MONCHAN has no meaning in priority mode.</p> <p><b>NOTE:</b> MONCHAN must only be changed when the Latency Monitor is in disabled mode.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                           |              |               |     |                         |                        |     |                            |                           |     |                     |                         |     |                  |          |     |                                |     |                             |     |                                |     |                   |
| 16–31<br>MAXLAT  | <p>Maximum Latency.</p> <p>This field specifies the maximum latency monitored, in microcycles, before a latency timeout. If MAXLAT=0, the latency keeps on being counted and measured, but the comparison with MAXLAT asserts neither the LTO1/2 fields in <a href="#">eTPU Module Configuration Register (eTPU_SCR_MCR)</a> nor issues a Global Exception. See <a href="#">Latency Monitor</a> for the detailed latency timeout description.</p> <p><b>NOTE:</b> MAXLAT must only be changed when the Latency Monitor is in disabled mode.</p>                                                                                                                                                                                                                                                                                                                                                  |                           |              |               |     |                         |                        |     |                            |                           |     |                     |                         |     |                  |          |     |                                |     |                             |     |                                |     |                   |

35.10.2 eTPU Latency Monitor Measure Register for Engine 1 (eTPU\_LMR\_LMnMR\_ENG1)

This register holds the latency measured value.

Address: 0h base + 84h offset + (8d × i), where i=0d to 5d

|       |         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |
|-------|---------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|
| Bit   | 0       | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15  |
| R     | 0       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |
| W     |         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |
| Reset | 0       | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   |
| Bit   | 16      | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31  |
| R     | LATENCY |    |    |    |    |    |    |    |    |    |    |    |    |    |    | LTC |
| W     |         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |
| Reset | 0       | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   |

eTPU\_LMR\_LMnMR\_ENG1 field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>Reserved | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 16–30<br>LATENCY | Latency measurement.<br><br>This read-only field holds the value of the greatest latency measured, in microcycles, since reset or the latest clear. When the value is greater than the maximum specified in <a href="#">MAXLAT</a> and MAXLAT value is not zero, a Global Exception is issued and <a href="#">LTO1</a> (engine 1) or <a href="#">LTO2</a> (engine 2) is set. See <a href="#">Latency Monitor</a> for the detailed latency measurement description.<br><br><b>NOTE:</b> When read, the LATENCY field is concatenated with the LTC bit (LSB) to form a 16-bit value.<br><b>NOTE:</b> Any latency measured above 0xFFFF is captured as 0xFFFF into the LATENCY+LTC field. |
| 31<br>LTC        | Latency measurement clear<br><br>Writing a 1 to the LTC bit clears the LATENCY field.<br><br><b>NOTE:</b> When read, the LATENCY field is concatenated with the LTC bit (LSB) to form a 16-bit value.<br><b>NOTE:</b> The LATENCY value is not automatically cleared when the Latency Monitor is disabled.<br><br>0   Keep the LATENCY field value.<br>1   Clear the LATENCY field value.                                                                                                                                                                                                                                                                                              |

### 35.10.3 eTPU Latency Monitor Configuration Register for Engine 2 (eTPU\_LMR\_LMnCR\_ENG2)

This register configures the operation of individual latency monitors.

Address: 0h base + C0h offset + (8d × i), where i=0d to 5d

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

#### eTPU\_LMR\_LMnCR\_ENG2 field descriptions

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                           |              |              |     |                         |                        |     |                            |                           |     |                     |                         |     |                  |          |     |                                |     |                             |     |                                |     |                   |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|--------------|--------------|-----|-------------------------|------------------------|-----|----------------------------|---------------------------|-----|---------------------|-------------------------|-----|------------------|----------|-----|--------------------------------|-----|-----------------------------|-----|--------------------------------|-----|-------------------|
| 0–1<br>LMM      | <p>Latency Monitor Mode.</p> <p>This field selects the operation mode of the latency monitor. See the Latency Monitor section for detailed mode descriptions.</p> <p><b>NOTE:</b> LMM must be in disabled mode to be changed to channel or priority mode.</p> <p>CPR must not be changed while LMM of any Latency Monitor is in priority mode.</p> <p>00 Disabled mode</p> <p>01 Channel mode</p> <p>10 Priority mode</p> <p>11 Reserved</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                           |              |              |     |                         |                        |     |                            |                           |     |                     |                         |     |                  |          |     |                                |     |                             |     |                                |     |                   |
| 2–4<br>Reserved | <p>Reserved.</p> <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                           |              |              |     |                         |                        |     |                            |                           |     |                     |                         |     |                  |          |     |                                |     |                             |     |                                |     |                   |
| 5–7<br>SRSEL    | <p>Service Request Selection</p> <p>This field selects which service requests are monitored. The SRSEL value has different meanings in channel and priority modes, according to the following table.</p> <p><b>Table 35-66. SRSEL - Service Request Selection</b></p> <table><tr><th>SRSEL</th><th>Channel Mode</th><th>Channel Mode</th></tr><tr><td>001</td><td>Link SRs monitored only</td><td>Low priority monitored</td></tr><tr><td>010</td><td>Channel SRs monitored only</td><td>Medium priority monitored</td></tr><tr><td>100</td><td>HSRs monitored only</td><td>High priority monitored</td></tr><tr><td>000</td><td>No SRs monitored</td><td rowspan="5">Reserved</td></tr><tr><td>011</td><td>Channel and Link SRs monitored</td></tr><tr><td>101</td><td>HSRs and Link SRs monitored</td></tr><tr><td>110</td><td>HSRs and Channel SRs monitored</td></tr><tr><td>111</td><td>All SRs monitored</td></tr></table> | SRSEL                     | Channel Mode | Channel Mode | 001 | Link SRs monitored only | Low priority monitored | 010 | Channel SRs monitored only | Medium priority monitored | 100 | HSRs monitored only | High priority monitored | 000 | No SRs monitored | Reserved | 011 | Channel and Link SRs monitored | 101 | HSRs and Link SRs monitored | 110 | HSRs and Channel SRs monitored | 111 | All SRs monitored |
| SRSEL           | Channel Mode                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Channel Mode              |              |              |     |                         |                        |     |                            |                           |     |                     |                         |     |                  |          |     |                                |     |                             |     |                                |     |                   |
| 001             | Link SRs monitored only                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Low priority monitored    |              |              |     |                         |                        |     |                            |                           |     |                     |                         |     |                  |          |     |                                |     |                             |     |                                |     |                   |
| 010             | Channel SRs monitored only                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Medium priority monitored |              |              |     |                         |                        |     |                            |                           |     |                     |                         |     |                  |          |     |                                |     |                             |     |                                |     |                   |
| 100             | HSRs monitored only                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | High priority monitored   |              |              |     |                         |                        |     |                            |                           |     |                     |                         |     |                  |          |     |                                |     |                             |     |                                |     |                   |
| 000             | No SRs monitored                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | Reserved                  |              |              |     |                         |                        |     |                            |                           |     |                     |                         |     |                  |          |     |                                |     |                             |     |                                |     |                   |
| 011             | Channel and Link SRs monitored                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                           |              |              |     |                         |                        |     |                            |                           |     |                     |                         |     |                  |          |     |                                |     |                             |     |                                |     |                   |
| 101             | HSRs and Link SRs monitored                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                           |              |              |     |                         |                        |     |                            |                           |     |                     |                         |     |                  |          |     |                                |     |                             |     |                                |     |                   |
| 110             | HSRs and Channel SRs monitored                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                           |              |              |     |                         |                        |     |                            |                           |     |                     |                         |     |                  |          |     |                                |     |                             |     |                                |     |                   |
| 111             | All SRs monitored                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                           |              |              |     |                         |                        |     |                            |                           |     |                     |                         |     |                  |          |     |                                |     |                             |     |                                |     |                   |

Table continues on the next page...

### eTPU\_LMR\_LMnCR\_ENG2 field descriptions (continued)

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 8–10<br>Reserved | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 11–15<br>MONCHAN | Monitored Channel.<br><br>When the latency monitor is configured for channel mode, this field selects which channel has the latency monitored. MONCHAN has no meaning in priority mode.<br><br><b>NOTE:</b> MONCHAN must only be changed when the Latency Monitor is in disabled mode.                                                                                                                                                                                                                                                   |
| 16–31<br>MAXLAT  | Maximum Latency.<br><br>This field specifies the maximum latency monitored, in microcycles, before a latency timeout. If MAXLAT=0, the latency keeps on being counted and measured, but the comparison with MAXLAT asserts neither the LTO1/2 fields in <a href="#">eTPU Module Configuration Register (eTPU_SCR_MCR)</a> nor issues a Global Exception. See <a href="#">Latency Monitor</a> for the detailed latency timeout description.<br><br><b>NOTE:</b> MAXLAT must only be changed when the Latency Monitor is in disabled mode. |

## 35.10.4 eTPU Latency Monitor Measure Register for Engine 2 (eTPU\_LMR\_LMnMR\_ENG2)

This register holds the latency measured value.

Address: 0h base + C4h offset + (8d × i), where i=0d to 5d

|       |         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|-------|---------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| Bit   | 0       | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R     | 0       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| W     |         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Reset | 0       | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Bit   | 16      | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| R     | LATENCY |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| W     |         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Reset | 0       | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

### eTPU\_LMR\_LMnMR\_ENG2 field descriptions

| Field            | Description                                                                                              |
|------------------|----------------------------------------------------------------------------------------------------------|
| 0–15<br>Reserved | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 16–30<br>LATENCY | Latency measurement.                                                                                     |

Table continues on the next page...

**eTPU\_LMR\_LMnMR\_ENG2 field descriptions (continued)**

| Field     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|           | <p>This read-only field holds the value of the greatest latency measured, in microcycles, since reset or the latest clear. When the value is greater than the maximum specified in <a href="#">MAXLAT</a> and MAXLAT value is not zero, a Global Exception is issued and <a href="#">LTO1</a> (engine 1) or <a href="#">LTO2</a> (engine 2) is set. See <a href="#">Latency Monitor</a> for the detailed latency measurement description.</p> <p><b>NOTE:</b> When read, the LATENCY field is concatenated with the LTC bit (LSB) to form a 16-bit value.</p> <p><b>NOTE:</b> Any latency measured above 0xFFFF is captured as 0xFFFF into the LATENCY+LTC field.</p> |
| 31<br>LTC | <p>Latency measurement clear.</p> <p>Writing a 1 to the LTC bit clears the LATENCY field.</p> <p><b>NOTE:</b> When read, the LATENCY field is concatenated with the LTC bit (LSB) to form a 16-bit value.</p> <p><b>NOTE:</b> The LATENCY value is not automatically cleared when the Latency Monitor is disabled.</p> <p>0 Keep the LATENCY field value.<br/>1 Clear the LATENCY field value.</p>                                                                                                                                                                                                                                                                    |

**35.11 Memory Error Support memory map and registers****eTPU\_MESR memory map**

| Address offset (hex) | Register name                                                        | Width (in bits) | Access | Reset value | Section/page                  |
|----------------------|----------------------------------------------------------------------|-----------------|--------|-------------|-------------------------------|
| 100                  | eTPU Memory Error Control Register (eTPU_MESR_MECR)                  | 32              | R/W    | 0000_0000h  | <a href="#">35.11.1/1346</a>  |
| 104                  | eTPU Data Error Injection Address Register (eTPU_MESR_DEIAR)         | 32              | R/W    | 0000_0000h  | <a href="#">35.11.2/1349</a>  |
| 108                  | eTPU Data Error Injection Data Pattern Register (eTPU_MESR_DEIDPR)   | 32              | R/W    | 0000_0000h  | <a href="#">35.11.3/1350</a>  |
| 10C                  | eTPU Data Error Injection Parity Pattern Register (eTPU_MESR_DEIPPR) | 32              | R/W    | 0000_0000h  | <a href="#">35.11.4/1350</a>  |
| 110                  | eTPU Data Error Report Address Register (eTPU_MESR_DERAR)            | 32              | R      | 0000_0000h  | <a href="#">35.11.5/1352</a>  |
| 114                  | eTPU Data Error Report Data Register (eTPU_MESR_DERDR)               | 32              | R      | 0000_0000h  | <a href="#">35.11.6/1353</a>  |
| 118                  | eTPU Data Error Report Syndrome Register (eTPU_MESR_DERSR)           | 32              | R      | 0000_0000h  | <a href="#">35.11.7/1354</a>  |
| 120                  | eTPU Memory Error Status Register (eTPU_MESR_MESR)                   | 32              | R/W    | 0000_0000h  | <a href="#">35.11.8/1356</a>  |
| 124                  | eTPU Code Error Injection Address Register (eTPU_MESR_CEIAR)         | 32              | R/W    | 0000_0000h  | <a href="#">35.11.9/1359</a>  |
| 128                  | eTPU Code Error Injection Data Pattern Register (eTPU_MESR_CEIDPR)   | 32              | R/W    | 0000_0000h  | <a href="#">35.11.10/1360</a> |
| 12C                  | eTPU Code Error Injection Parity Pattern Register (eTPU_MESR_CEIPPR) | 32              | R/W    | 0000_0000h  | <a href="#">35.11.11/1360</a> |

Table continues on the next page...

**eTPU\_MESR memory map (continued)**

| Address offset (hex) | Register name                                              | Width (in bits) | Access | Reset value | Section/ page                 |
|----------------------|------------------------------------------------------------|-----------------|--------|-------------|-------------------------------|
| 130                  | eTPU Code Error Report Address Register (eTPU_MESR_CERAR)  | 32              | R      | 0000_0000h  | <a href="#">35.11.12/1361</a> |
| 134                  | eTPU Code Error Report Data Register (eTPU_MESR_CERDR)     | 32              | R      | 0000_0000h  | <a href="#">35.11.13/1362</a> |
| 138                  | eTPU Code Error Report Syndrome Register (eTPU_MESR_CERSR) | 32              | R      | 0000_0000h  | <a href="#">35.11.14/1363</a> |
| 13C                  | eTPU Code Error Fix Register (eTPU_MESR_CEFR)              | 32              | R/W    | 0000_0000h  | <a href="#">35.11.15/1364</a> |

**35.11.1 eTPU Memory Error Control Register (eTPU\_MESR\_MECR)**

This register gathers control bits of both SCM and SDM error detection and correction support.

Address: 0h base + 100h offset = 100h

|       |     |   |   |   |   |   |   |   |   |   |    |    |    |    |    |       |
|-------|-----|---|---|---|---|---|---|---|---|---|----|----|----|----|----|-------|
| Bit   | 0   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15    |
| R     | 0   |   |   |   |   |   |   |   |   |   |    |    |    |    |    | MECIE |
| W     | FME |   |   |   |   |   |   |   |   |   |    |    |    |    |    |       |
| Reset | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0     |

|       |       |       |    |    |    |       |       |      |      |    |    |    |       |       |      |    |
|-------|-------|-------|----|----|----|-------|-------|------|------|----|----|----|-------|-------|------|----|
| Bit   | 16    | 17    | 18 | 19 | 20 | 21    | 22    | 23   | 24   | 25 | 26 | 27 | 28    | 29    | 30   | 31 |
| R     | HDEIE | MDEIE | 0  |    |    | DTEND | DRDIS | DEDD | CEIE | 0  |    |    | CTEND | CRDIS | CEDD |    |
| W     |       |       |    |    |    |       |       |      |      |    |    |    |       |       |      |    |
| Reset | 0     | 0     | 0  | 0  | 0  | 0     | 0     | 0    | 0    | 0  | 0  | 0  | 0     | 0     | 0    | 0  |

**eTPU\_MESR\_MECR field descriptions**

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                              |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>FME         | Field Modification Enable.<br><br>Negation of this bit disables the modification of all fields in the register, except MECIE, DRDIS, CRDIS and FME itself:<br><br>Write accesses do not cause a bus error, regardless of FME state.<br><br>0 Fields cannot be modified during a write access, except MECIE, DRDIS, CRDIS and FME.<br>1 All fields can be modified during a write access. |
| 1–14<br>Reserved | Reserved.                                                                                                                                                                                                                                                                                                                                                                                |

Table continues on the next page...

**eTPU\_MESR\_MECR field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 15<br>MECIE       | <p>Memory Error Correction Interrupt Enable.</p> <p>This bit enables the Memory Error Correction Interrupt (see <a href="#">Interrupt types and sources</a>).</p> <p>0 Memory Error Correction Interrupt disabled.</p> <p>1 Memory Error Correction Interrupt enabled.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 16<br>HDEIE       | <p>Host Data Error Injection Enable</p> <p>This bit enables error injection into the SDM host read path, including CDC transfers.</p> <p>The <a href="#">eTPU Data Error Injection Address Register (eTPU_MESR_DEIAR)</a> specifies the address where the error is injected, while <a href="#">eTPU Data Error Injection Data Pattern Register (eTPU_MESR_DEIDPR)</a> and <a href="#">eTPU Data Error Injection Parity Pattern Register (eTPU_MESR_DEIPPR)</a> define the bit flips on data and parity, respectively.</p> <p><b>NOTE:</b> Data error injection is also performed when data error detection is disabled (DEDD=1).</p> <p>0 SDM error injection disabled for host reads.</p> <p>1 SDM error injection enabled for host reads.</p>        |
| 17<br>MDEIE       | <p>Microengine(s) Data Error Injection Enable.</p> <p>This bit enables error injection into the SDM microengine read path.</p> <p>The <a href="#">eTPU Data Error Injection Address Register (eTPU_MESR_DEIAR)</a> specifies the address where the error is injected, while <a href="#">eTPU Data Error Injection Data Pattern Register (eTPU_MESR_DEIDPR)</a> and <a href="#">eTPU Data Error Injection Parity Pattern Register (eTPU_MESR_DEIPPR)</a> define the bit flips on data and parity, respectively.</p> <p><b>NOTE:</b> Data error injection is also performed when data error detection is disabled (DEDD=1).</p> <p>0 SDM error injection disabled for microengine reads.</p> <p>1 SDM error injection enabled for microengine reads.</p> |
| 18–20<br>Reserved | <p>Reserved.</p> <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 21<br>DTEND       | <p>Thread End on Data Non-correctable Error.</p> <p>This bit controls the behavior of the microengine when a non-correctable data error occurs. For more details, see <a href="#">Memory error support</a>.</p> <p><b>NOTE:</b> DTEND must only be changed when the engine is stopped or halted, or when all channels are disabled.</p> <p>0 Thread continues on non-correctable data read access by the microengine.</p> <p>1 Thread ends on non-correctable data read access by the microengine.</p>                                                                                                                                                                                                                                                 |
| 22<br>DRDIS       | <p>Data Error Report Disable.</p> <p>This bit disables the sampling of SDM error address, data and syndrome into the eTPU Memory Error Support Registers, and <a href="#">eTPU Data Error Report Syndrome Register (eTPU_MESR_DERSR)</a>. It does not, however, disable the setting of <a href="#">eTPU Memory Error Status Register (eTPU_MESR_MESR)</a> bits DCERR and DNCERR.</p>                                                                                                                                                                                                                                                                                                                                                                   |

Table continues on the next page...

**eTPU\_MESR\_MECR field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | <p><b>NOTE:</b> To guarantee coherent reading of <a href="#">eTPU Data Error Report Address Register (eTPU_MESR_DERAR)</a>, <a href="#">eTPU Data Error Report Data Register (eTPU_MESR_DERDR)</a>, and <a href="#">eTPU Data Error Report Syndrome Register (eTPU_MESR_DERSR)</a>, DRDIS must be asserted beforehand if SDM concurrent accesses can happen, by host or microengine. After the registers are read, DRDIS may be negated again to allow new samplings.</p> <p>0 SDM error report sampling enabled.</p> <p>1 SDM error report sampling disabled.</p>                                                                                                                                                                                                                 |
| 23<br>DEDD        | <p>Data Error Detection Disable.</p> <p>This bit disables SDM error detection.</p> <p>The disabled state turns ineffective any action due to error detection, including consequently error correction.</p> <p>0 SDM error detection enabled.</p> <p>1 SDM error detection disabled.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 24<br>CEIE        | <p>Code Error Injection Enable.</p> <p>This bit enables error injection into the SCM read path, either on microengine accesses (see <a href="#">VIS</a> =0) or host accesses (<a href="#">VIS</a> = 1)</p> <p><a href="#">eTPU Code Error Injection Address Register (eTPU_MESR_CEIAR)</a> specifies the address where the error is injected, while <a href="#">eTPU Code Error Injection Data Pattern Register (eTPU_MESR_CeidPR)</a> and <a href="#">eTPU Code Error Injection Parity Pattern Register (eTPU_MESR_CeiPPR)</a> define the bit flips on data and parity, respectively.</p> <p><b>NOTE:</b> Code error injection is also performed when code error detection is disabled (CEDD=1).</p> <p>0 SCM error injection disabled.</p> <p>1 SCM error injection enabled.</p> |
| 25–28<br>Reserved | <p>Reserved.</p> <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 29<br>CTEND       | <p>Thread End on Code Non-correctable Error.</p> <p>This bit controls the behavior of the microengine when a non-correctable code error occurs. For more details, see <a href="#">Memory error support</a>.</p> <p><b>NOTE:</b> CTEND must only be changed when the engine is stopped or halted, or when all channels are disabled.</p> <p>0 Thread continues on instruction fetch non-correctable error.</p> <p>1 Thread ends on instruction fetch non-correctable error.</p>                                                                                                                                                                                                                                                                                                     |
| 30<br>CRDIS       | <p>Code Error Report Disable.</p> <p>This bit disables the sampling of SCM error address, data and syndrome into the <a href="#">eTPU Code Error Report Address Register (eTPU_MESR_CERAR)</a>, <a href="#">eTPU Code Error Report Data Register (eTPU_MESR_CERDR)</a>, and <a href="#">eTPU Code Error Report Syndrome Register (eTPU_MESR_CERSR)</a>. It does not, however, disable the setting of CCERR and CNCERR in <a href="#">eTPU Memory Error Status Register (eTPU_MESR_MESR)</a>.</p>                                                                                                                                                                                                                                                                                   |

*Table continues on the next page...*



**eTPU\_MESR\_MECR field descriptions (continued)**

| Field      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|            | <p><b>NOTE:</b> To guarantee coherent reading of <a href="#">eTPU Code Error Report Address Register (eTPU_MESR_CERAR)</a>, <a href="#">eTPU Code Error Report Data Register (eTPU_MESR_CERDR)</a>, and <a href="#">eTPU Code Error Report Syndrome Register (eTPU_MESR_CERSR)</a>, CRDIS must be asserted beforehand if SCM concurrent accesses can happen, by host or microengine. After the registers are read, CRDIS may be negated again to allow new samplings.</p> <p>0 SCM error report sampling enabled.</p> <p>1 SCM error report sampling disabled.</p> |
| 31<br>CEDD | <p>Code Error Detection Disable.</p> <p>This bit disables SCM error detection.</p> <p>The disabled state turns ineffective any action due to error detection, including consequently error correction.</p> <p>0 SCM error detection enabled.</p> <p>1 SCM error detection disabled.</p>                                                                                                                                                                                                                                                                            |

**35.11.2 eTPU Data Error Injection Address Register (eTPU\_MESR\_DEIAR)**

This register holds the address where error is to be injected into the SDM read data path. The two least significant bits are 0 (read-only), so that the address is always word-aligned.

**NOTE**

This register must only be modified when [HDEIE](#) and [MDEIE](#) are both negated.

Address: 0h base + 104h offset = 104h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**eTPU\_MESR\_DEIAR field descriptions**

| Field             | Description                                                                                                                                                                                                                                                       |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–17<br>Reserved  | <p>Reserved.</p> <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                               |
| 18–29<br>INJ_ADDR | <p>Injection Address.</p> <p>INJ_ADDR holds the (word) address where an error is injected into the SDM read data path, as specified by <a href="#">eTPU Data Error Injection Data Pattern Register (eTPU_MESR_DEIDPR)</a> and <a href="#">eTPU Data Error</a></p> |

*Table continues on the next page...*

**eTPU\_MESR\_DEIAR field descriptions (continued)**

| Field             | Description                                                                                                                                           |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | <a href="#">Injection Parity Pattern Register (eTPU_MESR_DEIPPR)</a> . For more information on error injection, see <a href="#">Error injection</a> . |
| 30–31<br>Reserved | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                              |

**35.11.3 eTPU Data Error Injection Data Pattern Register (eTPU\_MESR\_DEIDPR)**

This register is used to specify the bits in which error is injected into the SDM read data path.

**NOTE**

This register must only be modified when [HDEIE](#) and [MDEIE](#) are both negated.

Address: 0h base + 108h offset = 108h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**eTPU\_MESR\_DEIDPR field descriptions**

| Field         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>DFLIP | <p>Data Flip bits.</p> <p>Each bit corresponds to a bit in the actual data read from the SDM address defined in <a href="#">eTPU Data Error Injection Address Register (eTPU_MESR_DEIAR)</a>, specifying whether it should be inverted or not. The inversion only occurs if the SDM error injection is enabled by HDEIE and/or MDEIE in <a href="#">eTPU Memory Error Control Register (eTPU_MESR_MECR)</a>. For more information on error injection, see <a href="#">Error injection</a>.</p> <p>0 Read data bit is not inverted.</p> <p>1 Read data bit is inverted.</p> |

**35.11.4 eTPU Data Error Injection Parity Pattern Register (eTPU\_MESR\_DEIPPR)**

This register is used to specify the bits in which error is injected into the SDM read parity path.

**NOTE**

This register must only be modified when [HDEIE](#) and [MDEIE](#) are both negated.

Address: 0h base + 10Ch offset = 10Ch

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   | 0 |   |    |    |    |    |    |    | 0  |    |    |    |    |    |    |    | 0  |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**eTPU\_MESR\_DEIPPR field descriptions**

| Field                | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–2<br>Reserved      | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 3–7<br>PFLIP_19_15   | Parity Flip bits.<br><br>Each bit corresponds to a bit in the actual parity read from the SDM address defined in <a href="#">eTPU Data Error Injection Address Register (eTPU_MESR_DEIAR)</a> , specifying whether it should be inverted or not. The inversion only occurs if the SDM error injection is enabled by the HDEIE and/or MDEIE fields in <a href="#">eTPU Memory Error Control Register (eTPU_MESR_MECCR)</a> . Bits PFLIP[19:15] corresponds to the parity bits for the most significant byte, PFLIP[14:10] to the next most significant byte and so on until PFLIP[4:0] for the least significant byte. For more information on error injection, see <a href="#">Error injection</a> .<br><br>0 Parity data bit is not inverted.<br>1 Parity data bit is inverted. |
| 8–10<br>Reserved     | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 11–15<br>PFLIP_14_10 | Parity Flip bits.<br><br>Each bit corresponds to a bit in the actual parity read from the SDM address defined in <a href="#">eTPU Data Error Injection Address Register (eTPU_MESR_DEIAR)</a> , specifying whether it should be inverted or not. The inversion only occurs if the SDM error injection is enabled by the HDEIE and/or MDEIE fields in <a href="#">eTPU Memory Error Control Register (eTPU_MESR_MECCR)</a> . Bits PFLIP[19:15] corresponds to the parity bits for the most significant byte, PFLIP[14:10] to the next most significant byte and so on until PFLIP[4:0] for the least significant byte. For more information on error injection, see <a href="#">Error injection</a> .<br><br>0 Parity data bit is not inverted.<br>1 Parity data bit is inverted. |
| 16–18<br>Reserved    | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 19–23<br>PFLIP_9_5   | Parity Flip bits.<br><br>Each bit corresponds to a bit in the actual parity read from the SDM address defined in <a href="#">eTPU Data Error Injection Address Register (eTPU_MESR_DEIAR)</a> , specifying whether it should be inverted or not. The inversion only occurs if the SDM error injection is enabled by the HDEIE and/or MDEIE fields in <a href="#">eTPU Memory Error Control Register (eTPU_MESR_MECCR)</a> . Bits PFLIP[19:15] corresponds to the parity bits for                                                                                                                                                                                                                                                                                                 |

Table continues on the next page...

**eTPU\_MESR\_DEIPPR field descriptions (continued)**

| Field              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                    | the most significant byte, PFLIP[14:10] to the next most significant byte and so on until PFLIP[4:0] for the least significant byte. For more information on error injection, see <a href="#">Error injection</a> .<br><br>0 Parity data bit is not inverted.<br>1 Parity data bit is inverted.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 24–26<br>Reserved  | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 27–31<br>PFLIP_4_0 | Parity Flip bits.<br><br>Each bit corresponds to a bit in the actual parity read from the SDM address defined in <a href="#">eTPU Data Error Injection Address Register (eTPU_MESR_DEIAR)</a> , specifying whether it should be inverted or not. The inversion only occurs if the SDM error injection is enabled by the HDEIE and/or MDEIE fields in <a href="#">eTPU Memory Error Control Register (eTPU_MESR_MECCR)</a> . Bits PFLIP[19:15] corresponds to the parity bits for the most significant byte, PFLIP[14:10] to the next most significant byte and so on until PFLIP[4:0] for the least significant byte. For more information on error injection, see <a href="#">Error injection</a> .<br><br>0 Parity data bit is not inverted.<br>1 Parity data bit is inverted. |

**35.11.5 eTPU Data Error Report Address Register (eTPU\_MESR\_DERAR)**

This register holds the address where the last non-correctable error was detected from the SDM. It also captures the address of the last correctable error detected, as long as [DNCERR](#) is 0. The two least significant bits are 0 (read-only), so that the address is always word-aligned.

Address: 0h base + 110h offset = 110h

|       |         |          |             |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---------|----------|-------------|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0       | 1        | 2           | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | ERR_ACC | 0        | ERR_CHANNEL |    |    |    |    |    | 0  |    |    |    |    |    |    |    |
| W     |         |          |             |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0       | 0        | 0           | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| Bit   | 16      | 17       | 18          | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0       | ERR_ADDR |             |    |    |    |    |    |    |    |    |    |    |    |    | 0  |
| W     |         |          |             |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0       | 0        | 0           | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**eTPU\_MESR\_DERAR field descriptions**

| Field              | Description                                                                                                                                                                                                                                                                                                                                                                                                 |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–1<br>ERR_ACC     | Error Access Source Indication.<br><br>This two-bit field indicates the source of the access when the error occurred.<br><br>00 Host - direct access<br>01 Engine 1<br>10 Engine 2<br>11 CDC                                                                                                                                                                                                                |
| 2<br>Reserved      | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                    |
| 3–7<br>ERR_CHANNEL | Error Channel Access Indication.<br><br>This field holds the number of the channel being serviced (not necessarily CHAN) when the error occurred. Its content is meaningful only when the field ERR_ACC indicates Engine 1 or Engine 2.                                                                                                                                                                     |
| 8–17<br>Reserved   | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                    |
| 18–29<br>ERR_ADDR  | Error Address.<br><br>ERR_ADDR holds the (word) address where the last captured SDM read error was detected. The last captured error is a non-correctable one if <b>DNCERR</b> =1, or a correctable one if <b>DNCERR</b> =0 and <b>DCERR</b> =1. The field is meaningless if both <b>DNCERR</b> and <b>DCERR</b> are 0. For more information on error detection, see <a href="#">Memory error support</a> . |
| 30–31<br>Reserved  | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                    |

**35.11.6 eTPU Data Error Report Data Register (eTPU\_MESR\_DERDR)**

This register holds the data read with the last non-correctable error detected from the SDM. It also captures the data of the last correctable error read detection, as long as **DNCERR** is 0. In both cases, the data is captured after the error correction logic. The data is meaningless if both **DNCERR** and **DCERR** are 0. For more information on error detection, see [Memory error support](#).

**NOTE**

Data is sampled before the error correction logic.

Address: 0h base + 114h offset = 114h

|       |      |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | DATA |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |      |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0    | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**eTPU\_MESR\_DERDR field descriptions**

| Field        | Description                                                                                                                                                                                      |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>DATA | DATA.<br><br><b>NOTE:</b> Individual data bytes are not meaningful when their respective BE bits in <a href="#">eTPU Data Error Report Syndrome Register (eTPU_MESR_DERSR)</a> are not asserted. |

**35.11.7 eTPU Data Error Report Syndrome Register (eTPU\_MESR\_DERSR)**

This register holds the syndrome read with the last non-correctable error detected from the SDM. It also captures the syndrome of the last correctable error read detection, as long as [DNCERR](#) is 0.

Address: 0h base + 118h offset = 118h

|       |     |    |    |    |    |    |    |    |     |    |    |    |    |    |    |    |
|-------|-----|----|----|----|----|----|----|----|-----|----|----|----|----|----|----|----|
| Bit   | 0   | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8   | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | BE3 | 0  |    |    |    |    |    |    | BE2 | 0  |    |    |    |    |    |    |
| W     |     |    |    |    |    |    |    |    |     |    |    |    |    |    |    |    |
| Reset | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| Bit   | 16  | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24  | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | BE1 | 0  |    |    |    |    |    |    | BE0 | 0  |    |    |    |    |    |    |
| W     |     |    |    |    |    |    |    |    |     |    |    |    |    |    |    |    |
| Reset | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**eTPU\_MESR\_DERSR field descriptions**

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>BE3        | Byte Enable 3.<br><br>Each BE bit corresponds to a report byte in a 32-bit aligned data word. BE3 corresponds to the most significant one. BE qualifies the error report for its respective byte.<br><br>0 Byte was not selected for read, therefore its respective error report fields are not meaningful.<br>1 Byte was selected for read, and the error report for the respective byte is meaningful. |
| 1–2<br>Reserved | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                 |
| 3–7<br>SYND3    | Error Syndrome 3.<br><br>SYND3 holds the syndrome of an error detected in the SDM read data path, as defined in <a href="#">Error Correction Code (ECC) and syndrome definition</a> . The field is meaningless if both <a href="#">DNCERR</a> and <a href="#">DCERR</a> are 0. For more information on error detection, see <a href="#">Memory error support</a> .                                       |

*Table continues on the next page...*

**eTPU\_MESR\_DERSR field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | <b>NOTE:</b> Syndrome fields which have their respective BE bits negated are not meaningful.                                                                                                                                                                                                                                                                                                                                                                              |
| 8<br>BE2          | <p>Byte Enable 2.</p> <p>Each BE bit corresponds to a report byte in a 32-bit aligned data word. BE3 corresponds to the most significant one. BE qualifies the error report for its respective byte.</p> <p>0 Byte was not selected for read, therefore its respective error report fields are not meaningful.</p> <p>1 Byte was selected for read, and the error report for the respective byte is meaningful.</p>                                                       |
| 9–10<br>Reserved  | <p>Reserved.</p> <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                       |
| 11–15<br>SYND2    | <p>Error Syndrome 2.</p> <p>SYND2 holds the syndrome of an error detected in the SDM read data path, as defined in the “Error Correction Code (ECC) and Syndrome Definition” section. The field is meaningless if both <a href="#">DNCERR</a> and <a href="#">DCERR</a> are 0. For more information on error detection, see <a href="#">Memory error support</a>.</p> <p><b>NOTE:</b> Syndrome fields which have their respective BE bits negated are not meaningful.</p> |
| 16<br>BE1         | <p>Byte Enable 1.</p> <p>Each BE bit corresponds to a report byte in a 32-bit aligned data word. BE3 corresponds to the most significant one. BE qualifies the error report for its respective byte.</p> <p>0 Byte was not selected for read, therefore its respective error report fields are not meaningful.</p> <p>1 Byte was selected for read, and the error report for the respective byte is meaningful.</p>                                                       |
| 17–18<br>Reserved | <p>Reserved.</p> <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                       |
| 19–23<br>SYND1    | <p>Error Syndrome 1.</p> <p>SYND1 holds the syndrome of an error detected in the SDM read data path, as defined in the “Error Correction Code (ECC) and Syndrome Definition” section. The field is meaningless if both <a href="#">DNCERR</a> and <a href="#">DCERR</a> are 0. For more information on error detection, see <a href="#">Memory error support</a>.</p> <p><b>NOTE:</b> Syndrome fields which have their respective BE bits negated are not meaningful.</p> |
| 24<br>BE0         | <p>Byte Enable 0.</p> <p>Each BE bit corresponds to a report byte in a 32-bit aligned data word. BE3 corresponds to the most significant one. BE qualifies the error report for its respective byte.</p> <p>0 Byte was not selected for read, therefore its respective error report fields are not meaningful.</p> <p>1 Byte was selected for read, and the error report for the respective byte is meaningful.</p>                                                       |
| 25–26<br>Reserved | <p>Reserved.</p> <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                       |
| 27–31<br>SYND0    | Error Syndrome 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

*Table continues on the next page...*

**eTPU\_MESR\_DERSR field descriptions (continued)**

| Field | Description                                                                                                                                                                                                                                                                                                          |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | SYND0 holds the syndrome of an error detected in the SDM read data path, as defined in the “Error Correction Code (ECC) and Syndrome Definition” section. The field is meaningless if both <b>DNCERR</b> and <b>DCERR</b> are 0. For more information on error detection, see <a href="#">Memory error support</a> . |
|       | <b>NOTE:</b> Syndrome fields which have their respective BE bits negated are not meaningful.                                                                                                                                                                                                                         |

**35.11.8 eTPU Memory Error Status Register (eTPU\_MESR\_MESR)**

This register gathers status bits of both SCM and SDM error detection and correction support.

Address: 0h base + 120h offset = 120h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**eTPU\_MESR\_MESR field descriptions**

| Field      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>DCERR | <p>Data Correctable Error flag.</p> <p>When read:</p> <p>0.....No SDM correctable error occurred.</p> <p>1.....An SDM correctable error occurred.</p> <p>Writing 1 to this bit clears the DCERR status flag.</p> <p>DCERR asserts even if DNCERR is asserted beforehand, but in this case the address and data of the erroneous access is not captured (see <a href="#">Memory Error Support memory map and registers</a> and following sections). DCERR is cleared by writing a 1 to it. Assertion of this bit can also assert the Memory Error Correction Interrupt (see <a href="#">Interrupt types and sources</a>).</p> <p><b>NOTE:</b> A correctable error is not considered for flagging if accompanied of a non-correctable one. Therefore, if both correctable and non-correctable data errors occur on the same access (in distinct bytes), the DNCERR flag asserts, but not the DCERR flag.</p> |

*Table continues on the next page...*



**eTPU\_MESR\_MESR field descriptions (continued)**

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                  | 0 Do not clear DCERR flag.<br>1 Clears DCERR flag                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 1–6<br>Reserved  | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 7<br>DCOVR       | Data Correctable Overrun flag.<br><br>Writing 1 to this bit clears the DCOVR status flag.<br><br>0 Data correctable error occurred while DCOVR = 1.<br>1 No data correctable error occurred while DCOVR = 1.<br><br>DCOVR asserts when an SDM correctable error occurs and the DCERR flag is already asserted. Assertion of this bit causes no interrupt.<br><br><b>NOTE:</b> A correctable error is not considered for flagging if accompanied by a non-correctable one. Therefore, if both correctable and non-correctable data errors occur on the same access (in distinct bytes), the DCOVR does not assert, even if DCERR is asserted. The DNCOVR flag, on the other hand, asserts if DNCERR is asserted. |
| 8<br>CCERR       | Code Error flag.<br><br>When read:<br>0.....No SCM correctable error occurred.<br>1.....An SCM correctable error occurred.<br><br>When written:<br>0.....Do not clear CCERR flag.<br>1.....Clears CCERR flag.<br><br>The CCERR flag asserts even if CNCERR is asserted beforehand, but in this case the address and data of the erroneous access is not captured (see <a href="#">eTPU Code Error Report Address Register (eTPU_MESR_CERAR)</a> and the following sections). CCERR is cleared by writing a 1 at the same position. Assertion of this bit can also assert the Memory Error Correction Interrupt (see <a href="#">Interrupt types and sources</a> ).                                              |
| 9–14<br>Reserved | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 15<br>CCOVR      | Code Correctable Overrun flag<br><br>This bit asserts when an SCM correctable error occurs and the CCERR flag is already asserted. Assertion of this bit causes no interrupt.<br><br>When read:<br>0.....No SCM correctable error overrun occurred.<br>1.....An SCM correctable error overrun occurred.<br><br>Writing 1 to this bit clears the CCOVR status flag:<br><br>0 Do not clear CCOVR flag.<br>1 Clear CCOVR flag.                                                                                                                                                                                                                                                                                     |

*Table continues on the next page...*

**eTPU\_MESR\_MESR field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 16<br>DNCERR      | <p>Data Non-Correctable Error flag</p> <p>DNCERR flags a non-correctable error.</p> <p>When read:</p> <p>0.....No SDM non-correctable error occurred.</p> <p>1.....An SDM non-correctable error occurred; registers eTPU Global Channel Registers are not updated by correctable error detections.</p> <p>DNCERR must be cleared (by writing a 1 to it) to allow correctable error updates to <a href="#">eTPU Data Error Report Address Register (eTPU_MESR_DERAR)</a> and <a href="#">eTPU Data Error Report Data Register (eTPU_MESR_DERDR)</a>.</p> <p>0 Do not clear DNCERR flag.</p> <p>1 Clear DNCERR flag.</p>                                                                                                                                                                                                                         |
| 17–22<br>Reserved | <p>Reserved.</p> <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 23<br>DNCOVR      | <p>Data Non-Correctable Overrun flag.</p> <p>DNCOVR asserts when an SDM non-correctable error occurs and the DNCERR flag is already asserted. Assertion of this bit causes no interrupt.</p> <p>When read:</p> <p>0.....No SDM non-correctable error overrun occurred.</p> <p>1.....An SDM non-correctable error overrun occurred.</p> <p>DNCOVR is cleared by writing a 1 to it.</p> <p>0 Do not clear DNCOVR flag.</p> <p>1 Clear DNCOVR flag.</p>                                                                                                                                                                                                                                                                                                                                                                                           |
| 24<br>CNCERR      | <p>Code Non-Correctable Error flag</p> <p>CNCERR flags a non-correctable error:</p> <p>When read:</p> <p>0.....An SCM non-correctable error occurred; the eTPU Global Channel Registers are not updated by correctable error detections.</p> <p>1.....an SCM non-correctable error occurred; <a href="#">eTPU Code Error Report Address Register (eTPU_MESR_CERAR)</a> and <a href="#">eTPU Code Error Report Data Register (eTPU_MESR_CERDR)</a> are not updated by correctable error detections.</p> <p>CNCERR must be cleared to allow correctable error updates to <a href="#">eTPU Code Error Report Address Register (eTPU_MESR_CERAR)</a> and <a href="#">eTPU Code Error Report Data Register (eTPU_MESR_CERDR)</a>.</p> <p>CNCERR is cleared by writing a 1 to it.</p> <p>0 Do not clear CNCERR flag.</p> <p>1 Clear CNCERR flag.</p> |
| 25–30<br>Reserved | <p>Reserved.</p> <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

*Table continues on the next page...*

**eTPU\_MESR\_MESR field descriptions (continued)**

| Field        | Description                                                                                                                                                                                                                                                                                                                                                                                                                |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31<br>CNCOVR | <p>Code Non-Correctable Overrun flag.</p> <p>CNCOVR asserts when an SCM non-correctable error occurs and the CNCERR flag is already asserted. Assertion of this bit causes no interrupt.</p> <p>When read:</p> <p>0.....an SCM non-correctable error overrun occurred.</p> <p>1.....An SCM non-correctable error overrun occurred.</p> <p>When written:</p> <p>0 Do not clear CNCOVR flag.</p> <p>1 Clear CNCOVR flag.</p> |

### 35.11.9 eTPU Code Error Injection Address Register (eTPU\_MESR\_CEIAR)

This register holds the address where error is to be injected into the SCM read data path. The two least significant bits are 0 (read-only), so that the address is always word-aligned.

#### NOTE

This register must only be modified when [CEIE](#) is negated.

Address: 0h base + 124h offset = 124h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16       | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | INJ_ADDR |    |    |    |    |    |    |    |    |    |    |    |    | 0  |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**eTPU\_MESR\_CEIAR field descriptions**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                           |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>Reserved  | <p>Reserved.</p> <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                   |
| 16–29<br>INJ_ADDR | <p>Injection Address.</p> <p>INJ_ADDR holds the (word) address where an error is injected in the SCM read data path, as specified by <a href="#">eTPU Code Error Injection Data Pattern Register (eTPU_MESR_CEI DPR)</a> and <a href="#">eTPU Code Error Injection Parity Pattern Register (eTPU_MESR_CEI PPR)</a>. For more information on error injection, see <a href="#">Error injection</a>.</p> |
| 30–31<br>Reserved | <p>Reserved.</p> <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                   |

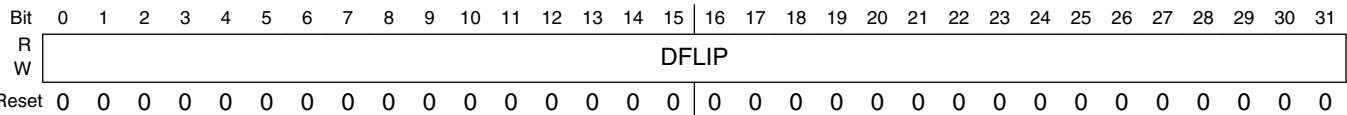
35.11.10 eTPU Code Error Injection Data Pattern Register (eTPU\_MESR\_CEIDPR)

This register is used to specify the bits in which error is injected into the SCM read data path.

NOTE

This register must only be modified when CEIE is negated.

Address: 0h base + 128h offset = 128h



eTPU\_MESR\_CEIDPR field descriptions

| Field         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>DFLIP | <p>Data Flip bits.</p> <p>Each bit corresponds to a bit in the actual data read from the SCM address defined in <a href="#">eTPU Code Error Injection Address Register (eTPU_MESR_CEIAR)</a>, specifying whether it should be inverted or not. The inversion only occurs if the SCM error injection is enabled by <a href="#">CEIE</a>. For more information on error injection, see <a href="#">Error injection</a>.</p> <p>0 Read data bit is not inverted.</p> <p>1 Read data bit is inverted.</p> |

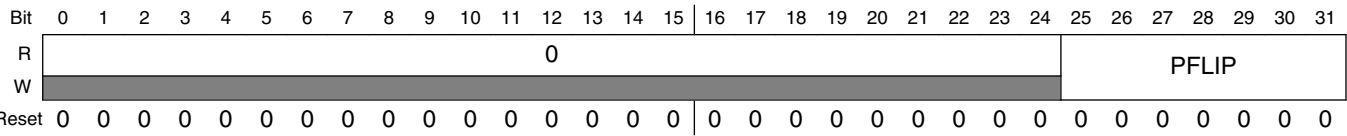
35.11.11 eTPU Code Error Injection Parity Pattern Register (eTPU\_MESR\_CEIPPR)

This register is used to specify the bits in which error is injected into the SCM read parity path.

NOTE

This register must only be modified when CEIE is negated.

Address: 0h base + 12Ch offset = 12Ch



**eTPU\_MESR\_CEIPPR field descriptions**

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–24<br>Reserved | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                   |
| 25–31<br>PFLIP   | Parity Flip bits.<br><br>Each bit corresponds to a bit in the actual parity/code read from the SCM address defined in <a href="#">eTPU Code Error Injection Address Register (eTPU_MESR_CEIAR)</a> , specifying whether it should be inverted or not. The inversion only occurs if the SCM error injection is enabled by <a href="#">CEIE</a> . For more information on error injection, see <a href="#">Error injection</a> .<br><br>0 Parity data bit is not inverted.<br>1 Parity data bit is inverted. |

**35.11.12 eTPU Code Error Report Address Register (eTPU\_MESR\_CERAR)**

This register holds the address where the last non-correctable error was detected from the SCM. It also captures the address of the last correctable error detected, as long as [CNCERR](#) is 0. The two least significant bits are 0 (read-only), so that the address is always word-aligned.

Address: 0h base + 130h offset = 130h

|       |         |   |   |             |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|---------|---|---|-------------|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0       | 1 | 2 | 3           | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | ERR_ACC |   | 0 | ERR_CHANNEL |   |   |   | 0 |   |   |    |    |    |    |    |    |
| W     |         |   |   |             |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0       | 0 | 0 | 0           | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 16       | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | ERR_ADDR |    |    |    |    |    |    |    |    |    |    |    |    |    | 0  |    |
| W     |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**eTPU\_MESR\_CERAR field descriptions**

| Field          | Description                                                                                                                                         |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–1<br>ERR_ACC | Error Access Source Indication.<br><br>This two-bit field indicates the source of the access when the error occurred.<br><br>00 Host<br>01 Engine 1 |

*Table continues on the next page...*

**eTPU\_MESR\_CERAR field descriptions (continued)**

| Field                  | Description                                                                                                                                                                                                                                                                                                                                                                                                    |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 10 Engine 2<br>11 MISC |                                                                                                                                                                                                                                                                                                                                                                                                                |
| 2<br>Reserved          | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                       |
| 3–7<br>ERR_CHANNEL     | Error Channel Access Indication.<br><br>This field holds the number of the channel being serviced (not necessarily CHAN) when the error occurred. Its content is meaningful only when the field ERR_ACC indicates Engine 1 or Engine 2.                                                                                                                                                                        |
| 8–15<br>Reserved       | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                       |
| 16–29<br>ERR_ADDR      | Error Address.<br><br>ERR_ADDR holds the (word) address where the last captured SCM read error was detected. The last captured error is a non-correctable one if <b>CNCERR</b> = 1, or a correctable one if <b>CNCERR</b> = 0 and <b>CCERR</b> = 1. The field is meaningless if both <b>CNCERR</b> and <b>CCERR</b> are 0. For more information on error detection, see <a href="#">Memory error support</a> . |
| 30–31<br>Reserved      | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                       |

**35.11.13 eTPU Code Error Report Data Register (eTPU\_MESR\_CERDR)**

This register holds the data read with the last non-correctable error detected from the SCM. It also captures the data of the last correctable error read detection, as long as **CNCERR** is 0. In both cases, the data is captured after the error correction logic. The data may be meaningless if both **CNCERR** and **CCERR** are 0. For more information on error detection, see [Memory error support](#).

**NOTE**

CODE is sampled before the error correction logic.

Address: 0h base + 134h offset = 134h

|       |      |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | CODE |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |      |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0    | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**eTPU\_MESR\_CERDR field descriptions**

| Field        | Description |
|--------------|-------------|
| 0–31<br>CODE | CODE[31-0]. |

### 35.11.14 eTPU Code Error Report Syndrome Register (eTPU\_MESR\_CERSR)

This register holds the syndrome and global parity read with the last non-correctable error detected from the SCM. It also captures the syndrome of the last correctable error read detection, as long as [CNCERR](#) is 0.

Address: 0h base + 138h offset = 138h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24   | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    | SYND |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0  | 0  | 0  | 0  | 0  | 0  |    |

**eTPU\_MESR\_CERSR field descriptions**

| Field            | Description                                                                                                                                                                                                                                                                                                                                                     |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–24<br>Reserved | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                        |
| 25–31<br>SYND    | Error Syndrome.<br><br>SYND holds the syndrome of an error detected in the SCM read data path, as defined in <a href="#">Error Correction Code (ECC) and syndrome definition</a> . The field is meaningless if both <a href="#">CNCERR</a> and <a href="#">CCERR</a> are 0. For more information on error detection, see <a href="#">Memory error support</a> . |

### 35.11.15 eTPU Code Error Fix Register (eTPU\_MESR\_CEFR)

This register is used to fix single-bit error in a specified SCM position. The fix is done automatically, concurrently with code execution, by the MISC logic (see [Error fixing](#)).

Address: 0h base + 13Ch offset = 13Ch

|       |           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |   |  |
|-------|-----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--|--|--|--|--|--|--|--|--|--|--|--|--|---|--|
| Bit   | 0         | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |  |  |  |  |  |  |  |  |  |  |  |  |  |   |  |
| R     | CFIXM     |    | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |   |  |
| W     |           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |   |  |
| Reset | 0         | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |  |  |  |  |  |  |  |  |  |  |  |  |  |   |  |
| Bit   | 16        | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |  |  |  |  |  |  |  |  |  |  |  |  |  |   |  |
| R     | CFIX_ADDR |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  | 0 |  |
| W     |           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |   |  |
| Reset | 0         | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |  |  |  |  |  |  |  |  |  |  |  |  |  |   |  |

#### eTPU\_MESR\_CEFR field descriptions

| Field              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                                                                                                                                                 |      |             |    |          |                         |    |          |                                   |    |          |                                   |    |           |                                                                                                                                                                 |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|------|-------------|----|----------|-------------------------|----|----------|-----------------------------------|----|----------|-----------------------------------|----|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–1<br>CFIXM       | <p>Code Fix Mode.</p> <p>This field specifies the SCM MISC fix logic operation mode as shown in the table below.</p> <p><b>NOTE:</b> The CFIXM field can only be changed to/from disabled mode.</p> <p><b>Table 35-100. CFIXM - Code Fix Mode</b></p> <table><tr><th>Value</th><th>Mode</th><th>Description</th></tr><tr><td>00</td><td>Disabled</td><td>SCM error fix disabled.</td></tr><tr><td>01</td><td>Reserved</td><td>Reserved value, must not be used.</td></tr><tr><td>10</td><td>Reserved</td><td>Reserved value, must not be used.</td></tr><tr><td>11</td><td>Automatic</td><td>The MISC logic rewrites the code word value read (already fixed by the error correction mechanism) into the same address whenever it finds a correctable error.</td></tr></table> | Value                                                                                                                                                           | Mode | Description | 00 | Disabled | SCM error fix disabled. | 01 | Reserved | Reserved value, must not be used. | 10 | Reserved | Reserved value, must not be used. | 11 | Automatic | The MISC logic rewrites the code word value read (already fixed by the error correction mechanism) into the same address whenever it finds a correctable error. |
| Value              | Mode                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Description                                                                                                                                                     |      |             |    |          |                         |    |          |                                   |    |          |                                   |    |           |                                                                                                                                                                 |
| 00                 | Disabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | SCM error fix disabled.                                                                                                                                         |      |             |    |          |                         |    |          |                                   |    |          |                                   |    |           |                                                                                                                                                                 |
| 01                 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Reserved value, must not be used.                                                                                                                               |      |             |    |          |                         |    |          |                                   |    |          |                                   |    |           |                                                                                                                                                                 |
| 10                 | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Reserved value, must not be used.                                                                                                                               |      |             |    |          |                         |    |          |                                   |    |          |                                   |    |           |                                                                                                                                                                 |
| 11                 | Automatic                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | The MISC logic rewrites the code word value read (already fixed by the error correction mechanism) into the same address whenever it finds a correctable error. |      |             |    |          |                         |    |          |                                   |    |          |                                   |    |           |                                                                                                                                                                 |
| 2–15<br>Reserved   | <p>Reserved.</p> <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                 |      |             |    |          |                         |    |          |                                   |    |          |                                   |    |           |                                                                                                                                                                 |
| 16–29<br>CFIX_ADDR | <p>Code Fix Address.</p> <p>This field is used for hardware debug purposes. CFIX_ADDR may change its contents and must not be written in automatic mode. For more information on error detection and correction, see <a href="#">Memory error support</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                                                                                                                                                                 |      |             |    |          |                         |    |          |                                   |    |          |                                   |    |           |                                                                                                                                                                 |
| 30–31<br>Reserved  | <p>Reserved.</p> <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                 |      |             |    |          |                         |    |          |                                   |    |          |                                   |    |           |                                                                                                                                                                 |



## 35.12 Global Channel memory map and registers

The registers in this section group, by type, the interrupt status and enable bits from all the channels. This organization eases management of all channels or groups of channels by a single interrupt handler routine. These bits, except the service and watchdog status, are mirrored in the individual channel registers, grouped by channel.

**eTPU\_GCR memory map**

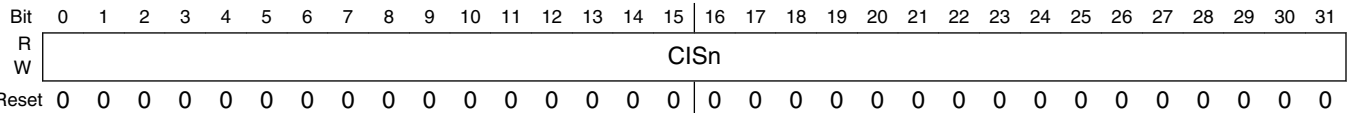
| Address offset (hex) | Register name                                                                                    | Width (in bits) | Access | Reset value | Section/ page                 |
|----------------------|--------------------------------------------------------------------------------------------------|-----------------|--------|-------------|-------------------------------|
| 200                  | eTPU Channel Interrupt Status Register for Engine 1 (eTPU_GCR_CISR_ENG1)                         | 32              | R/W    | 0000_0000h  | <a href="#">35.12.1/1366</a>  |
| 204                  | eTPU Channel Interrupt Status Register for Engine 2 (eTPU_GCR_CISR_ENG2)                         | 32              | R/W    | 0000_0000h  | <a href="#">35.12.2/1366</a>  |
| 210                  | eTPU Channel Data Transfer Request Status Register for Engine 1 (eTPU_GCR_CDTRSR_ENG1)           | 32              | R/W    | 0000_0000h  | <a href="#">35.12.3/1367</a>  |
| 214                  | eTPU Channel Data Transfer Request Status Register for Engine 2 (eTPU_GCR_CDTRSR_ENG2)           | 32              | R/W    | 0000_0000h  | <a href="#">35.12.4/1368</a>  |
| 220                  | eTPU Channel Interrupt Overflow Status Register for Engine 1 (eTPU_GCR_CIOSR_ENG1)               | 32              | R/W    | 0000_0000h  | <a href="#">35.12.5/1368</a>  |
| 224                  | eTPU Channel Interrupt Overflow Status Register for Engine 2 (eTPU_GCR_CIOSR_ENG2)               | 32              | R/W    | 0000_0000h  | <a href="#">35.12.6/1369</a>  |
| 230                  | eTPU Channel Data Transfer Request Overflow Status Register for Engine 1 (eTPU_GCR_CDTROSR_ENG1) | 32              | R/W    | 0000_0000h  | <a href="#">35.12.7/1370</a>  |
| 234                  | eTPU Channel Data Transfer Request Overflow Status Register for Engine 2 (eTPU_GCR_CDTROSR_ENG2) | 32              | R/W    | 0000_0000h  | <a href="#">35.12.8/1370</a>  |
| 240                  | eTPU Channel Interrupt Enable Register for Engine 1 (eTPU_GCR_CIER_ENG1)                         | 32              | R/W    | 0000_0000h  | <a href="#">35.12.9/1371</a>  |
| 244                  | eTPU Channel Interrupt Enable Register for Engine 2 (eTPU_GCR_CIER_ENG2)                         | 32              | R/W    | 0000_0000h  | <a href="#">35.12.10/1371</a> |
| 250                  | eTPU Channel Data Transfer Request Enable Register for Engine 1 (eTPU_GCR_CDTREER_ENG1)          | 32              | R/W    | 0000_0000h  | <a href="#">35.12.11/1372</a> |
| 254                  | eTPU Channel Data Transfer Request Enable Register for Engine 2 (eTPU_GCR_CDTREER_ENG2)          | 32              | R/W    | 0000_0000h  | <a href="#">35.12.12/1372</a> |
| 260                  | eTPU Watchdog Status Register for Engine 1 (eTPU_GCR_WDSR_ENG1)                                  | 32              | R/W    | 0000_0000h  | <a href="#">35.12.13/1373</a> |
| 264                  | eTPU Watchdog Status Register for Engine 2 (eTPU_GCR_WDSR_ENG2)                                  | 32              | R/W    | 0000_0000h  | <a href="#">35.12.14/1374</a> |
| 280                  | eTPU Channel Pending Service Status Register for Engine 1 (eTPU_GCR_CPSSR_ENG1)                  | 32              | R      | 0000_0000h  | <a href="#">35.12.15/1374</a> |
| 284                  | eTPU Channel Pending Service Status Register for Engine 2 (eTPU_GCR_CPSSR_ENG2)                  | 32              | R      | 0000_0000h  | <a href="#">35.12.16/1375</a> |
| 290                  | eTPU Channel Service Status Register for Engine 1 (eTPU_GCR_CSSR_ENG1)                           | 32              | R      | 0000_0000h  | <a href="#">35.12.17/1376</a> |
| 294                  | eTPU Channel Service Status Register for Engine 2 (eTPU_GCR_CSSR_ENG2)                           | 32              | R      | 0000_0000h  | <a href="#">35.12.18/1376</a> |

35.12.1 eTPU Channel Interrupt Status Register for Engine 1 (eTPU\_GCR\_CISR\_ENG1)

This register groups the host interrupt status from all channels. These bits are mirrored from the Channel Status Control register (see [eTPU Channel n Status Control Register for Engine 1 \(eTPU\\_CCCR\\_CnSCR\\_ENG1\)](#)) and the Host must write 1 to clear a status bit.

For details about interrupts, see [Interrupts and data transfer requests](#).

Address: 0h base + 200h offset = 200h



eTPU\_GCR\_CISR\_ENG1 field descriptions

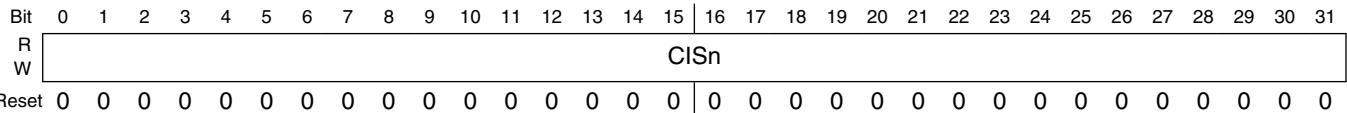
| Field        | Description                                                                                                                                                                                                                                                                                                          |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>CISn | Channel n Interrupt Status.<br><br>When read, each bit:<br><br>0.....Indicates that channel n has no pending interrupt to the CPU.<br>1.....Indicates that channel n has a pending interrupt to the CPU.<br><br>When written:<br><br>0.....Keep interrupt status bit unaltered.<br>1.....Clear interrupt status bit. |

35.12.2 eTPU Channel Interrupt Status Register for Engine 2 (eTPU\_GCR\_CISR\_ENG2)

This register groups the host interrupt status from all channels. These bits are mirrored from the Channel Status Control register (see [eTPU Channel n Status Control Register for Engine 2 \(eTPU\\_CCCR\\_CnSCR\\_ENG2\)](#)) and the Host must write 1 to clear a status bit.

For details about interrupts, see [Interrupts and data transfer requests](#).

Address: 0h base + 204h offset = 204h



**eTPU\_GCR\_CISR\_ENG2 field descriptions**

| Field        | Description                                                                                                                                                                                                                                                                                                                         |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>CISn | <p>Channel n Interrupt Status.</p> <p>When read, each bit:</p> <p>0.....Indicates that channel n has no pending interrupt to the CPU.</p> <p>1.....Indicates that channel n has a pending interrupt to the CPU.</p> <p>When written:</p> <p>0.....Keep interrupt status bit unaltered.</p> <p>1.....Clear interrupt status bit.</p> |

**35.12.3 eTPU Channel Data Transfer Request Status Register for Engine 1 (eTPU\_GCR\_CDTRSR\_ENG1)**

This register groups data transfer request status from all channels. Their bits are mirrored from the Channel Status Control registers (see [eTPU Channel n Status Control Register for Engine 1 \(eTPU\\_CCCR\\_CnSCR\\_ENG1\)](#)) and Host must write 1 to clear a status bit.

For details about interrupts, see [Interrupts and data transfer requests](#).

Address: 0h base + 210h offset = 210h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**eTPU\_GCR\_CDTRSR\_ENG1 field descriptions**

| Field         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>DTRSn | <p>Channel n Data Transfer Request Status.</p> <p>These bits mimic the corresponding eTPU DMA requests. DTRSn can be cleared by software writing a 1 to it or by the assertion of corresponding DMA completion acknowledge line.</p> <p>When read:</p> <p>0.....Indicates that channel n has no pending data transfer request.</p> <p>1.....Indicates that channel x has a pending data transfer request.</p> <p>When written:</p> <p>0.....Keep status bit unaltered.</p> <p>1.....Clear status bit.</p> |

### 35.12.4 eTPU Channel Data Transfer Request Status Register for Engine 2 (eTPU\_GCR\_CDTRSR\_ENG2)

This register groups data transfer request status from all channels. These bits are mirrored from the Channel Status Control registers (see [eTPU Channel n Status Control Register for Engine 2 \(eTPU\\_CCCR\\_CnSCR\\_ENG2\)](#)) and Host must write 1 to clear a status bit.

For details about interrupts, see [Interrupts and data transfer requests](#).

Address: 0h base + 214h offset = 214h

|       |       |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0     | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | DTRSn |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |       |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0     | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

#### eTPU\_GCR\_CDTRSR\_ENG2 field descriptions

| Field         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>DTRSn | <p>Channel n Data Transfer Request Status.</p> <p>These bits mimic the corresponding eTPU DMA requests. DTRSn can be cleared by software writing a 1 to it or by the assertion of corresponding DMA completion acknowledge line.</p> <p>When read:</p> <p>0.....Indicates that channel n has no pending data transfer request.</p> <p>1.....Indicates that channel x has a pending data transfer request.</p> <p>When written:</p> <p>0.....Keep status bit unaltered.</p> <p>1.....Clear status bit.</p> |

### 35.12.5 eTPU Channel Interrupt Overflow Status Register for Engine 1 (eTPU\_GCR\_CIOSR\_ENG1)

This register groups the interrupt overflow status from all channels. These bits are mirrored from the Channel Status Control registers (see [eTPU Channel n Status Control Register for Engine 1 \(eTPU\\_CCCR\\_CnSCR\\_ENG1\)](#)) and the Host must write 1 to clear a status bit.

For details about interrupts, see [Interrupts and data transfer requests](#).

Address: 0h base + 220h offset = 220h

|       |       |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0     | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | CIOSn |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |       |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0     | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**eTPU\_GCR\_CIOSR\_ENG1 field descriptions**

| Field         | Description                                                                                                                                                                                                                                                                                                 |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>CIOSn | <p>Channel n Interrupt Overflow Status</p> <p>When read:</p> <p>0.....Indicates that no interrupt overflow occurred in the channel.</p> <p>1.....Indicates that interrupt overflow occurred in the channel.</p> <p>When written:</p> <p>0.....Keep status bit unaltered.</p> <p>1.....Clear status bit.</p> |

**35.12.6 eTPU Channel Interrupt Overflow Status Register for Engine 2 (eTPU\_GCR\_CIOSR\_ENG2)**

This register groups the interrupt overflow status from all channels. These bits are mirrored from the Channel Status Control registers (see [eTPU Channel n Status Control Register for Engine 2 \(eTPU\\_CCCR\\_CnSCR\\_ENG2\)](#)) and the Host must write 1 to clear a status bit.

For details about interrupts, see [Interrupts and data transfer requests](#).

Address: 0h base + 224h offset = 224h

|       |                   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0                 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | <div>CIOStn</div> |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |                   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0                 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**eTPU\_GCR\_CIOSR\_ENG2 field descriptions**

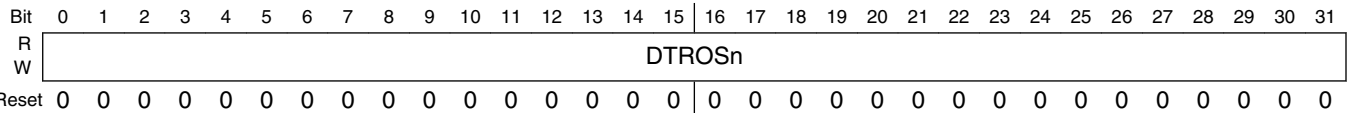
| Field         | Description                                                                                                                                                                                                                                                                                                 |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>CIOSn | <p>Channel n Interrupt Overflow Status</p> <p>When read:</p> <p>0.....Indicates that no interrupt overflow occurred in the channel.</p> <p>1.....Indicates that interrupt overflow occurred in the channel.</p> <p>When written:</p> <p>0.....Keep status bit unaltered.</p> <p>1.....Clear status bit.</p> |

35.12.7 eTPU Channel Data Transfer Request Overflow Status Register for Engine 1 (eTPU\_GCR\_CDTROSR\_ENG1)

This register groups the data transfer request overflow status from all channels. These bits are mirrored from the Channel Status Control registers (see [eTPU Channel n Status Control Register for Engine 1 \(eTPU\\_CCCR\\_CnSCR\\_ENG1\)](#)) and the CPU must write 1 to clear a status bit.

For details about interrupts, see [Interrupts and data transfer requests](#).

Address: 0h base + 230h offset = 230h



eTPU\_GCR\_CDTROSR\_ENG1 field descriptions

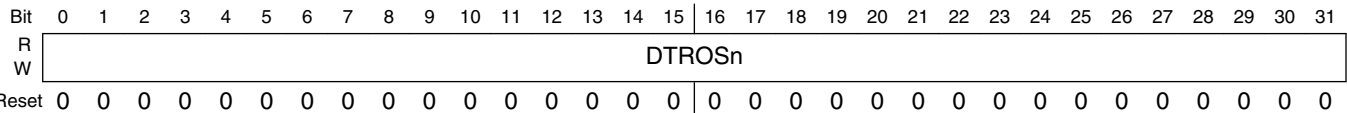
| Field          | Description                                                                                                                                                                                                                                                                                                               |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>DTROSn | Channel n Data Transfer Request Overflow Status.<br><br>When read:<br>0.....Indicates that no data transfer request overflow occurred in the channel.<br>1.....Indicates that data transfer request overflow occurred in the channel.<br><br>When written:<br>0.....Keep status bit unaltered.<br>1.....Clear status bit. |

35.12.8 eTPU Channel Data Transfer Request Overflow Status Register for Engine 2 (eTPU\_GCR\_CDTROSR\_ENG2)

This register groups the data transfer request overflow status from all channels. These bits are mirrored from the Channel Status Control registers (see [eTPU Channel n Status Control Register for Engine 2 \(eTPU\\_CCCR\\_CnSCR\\_ENG2\)](#)) and the CPU must write 1 to clear a status bit.

For details about interrupts, see [Interrupts and data transfer requests](#).

Address: 0h base + 234h offset = 234h



**eTPU\_GCR\_CDTROSR\_ENG2 field descriptions**

| Field          | Description                                                                                                                                                                                                                                                                                                               |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>DTROSn | Channel n Data Transfer Request Overflow Status.<br><br>When read:<br>0.....Indicates that no data transfer request overflow occurred in the channel.<br>1.....Indicates that data transfer request overflow occurred in the channel.<br><br>When written:<br>0.....Keep status bit unaltered.<br>1.....Clear status bit. |

**35.12.9 eTPU Channel Interrupt Enable Register for Engine 1 (eTPU\_GCR\_CIER\_ENG1)**

This register groups the data transfer request overflow status from all channels. These bits are mirrored from the Channel Status Control registers (see [eTPU Channel n Status Control Register for Engine 1 \(eTPU\\_CCCR\\_CnSCR\\_ENG1\)](#)).

For details about interrupts, see [Interrupts and data transfer requests](#).

Address: 0h base + 240h offset = 240h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**eTPU\_GCR\_CIER\_ENG1 field descriptions**

| Field        | Description                                                                                                      |
|--------------|------------------------------------------------------------------------------------------------------------------|
| 0–31<br>CIEn | Channel n Interrupt Enable.<br><br>0   Interrupt disabled for channel n.<br>1   Interrupt enabled for channel n. |

**35.12.10 eTPU Channel Interrupt Enable Register for Engine 2 (eTPU\_GCR\_CIER\_ENG2)**

This register groups the data transfer request overflow status from all channels. These bits are mirrored from the Channel Status Control registers (see [eTPU Channel n Status Control Register for Engine 2 \(eTPU\\_CCCR\\_CnSCR\\_ENG2\)](#)).

For details about interrupts, see [Interrupts and data transfer requests](#).

## Global Channel memory map and registers

Address: 0h base + 244h offset = 244h

| Bit   | 0    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|-------|------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| R     | CIEn |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |      |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0    | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

### eTPU\_GCR\_CIER\_ENG2 field descriptions

| Field                    | Description                                                                                                  |
|--------------------------|--------------------------------------------------------------------------------------------------------------|
| 0–31<br>CIE <sub>n</sub> | Channel n Interrupt Enable.<br><br>0 Interrupt disabled for channel n.<br>1 Interrupt enabled for channel n. |

## 35.12.11 eTPU Channel Data Transfer Request Enable Register for Engine 1 (eTPU\_GCR\_CDTRER\_ENG1)

This register groups the data transfer request enable from all channels. These bits are mirrored from the Channel Status Control registers (see [eTPU Channel n Status Control Register for Engine 1 \(eTPU\\_CCCR\\_CnSCR\\_ENG1\)](#)).

For details about interrupts, see [Interrupts and data transfer requests](#).

Address: 0h base + 250h offset = 250h

| Bit   | 0     | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|-------|-------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| R     | DTREn |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |       |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0     | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

### eTPU\_GCR\_CDTRER\_ENG1 field descriptions

| Field                     | Description                                                                                                                                      |
|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>DTRE <sub>n</sub> | Channel n Data Transfer Request Enable.<br><br>0 Data Transfer request disabled for channel n.<br>1 Data Transfer request enabled for channel n. |

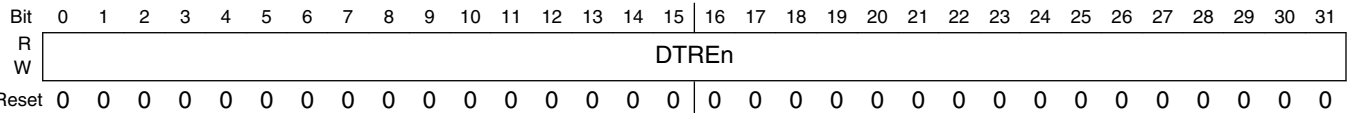
## 35.12.12 eTPU Channel Data Transfer Request Enable Register for Engine 2 (eTPU\_GCR\_CDTRER\_ENG2)

This register groups the data transfer request enable from all channels. These bits are mirrored from the Channel Status Control registers (see [eTPU Channel n Status Control Register for Engine 2 \(eTPU\\_CCCR\\_CnSCR\\_ENG2\)](#)).

For details about interrupts, see [Interrupts and data transfer requests](#).



Address: 0h base + 254h offset = 254h



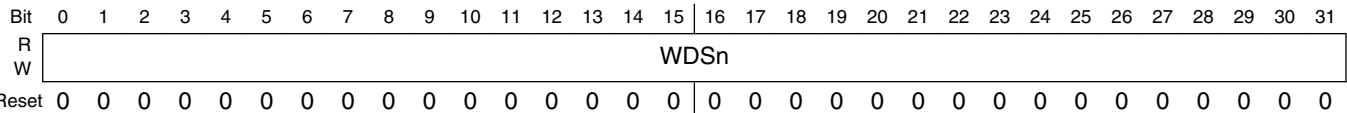
eTPU\_GCR\_CDTRER\_ENG2 field descriptions

| Field         | Description                                                                                                                                      |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>DTREn | Channel n Data Transfer Request Enable.<br><br>0 Data Transfer request disabled for channel n.<br>1 Data Transfer request enabled for channel n. |

35.12.13 eTPU Watchdog Status Register for Engine 1 (eTPU\_GCR\_WDSR\_ENG1)

This register indicates the watchdog influence on each of the engine channels.

Address: 0h base + 260h offset = 260h



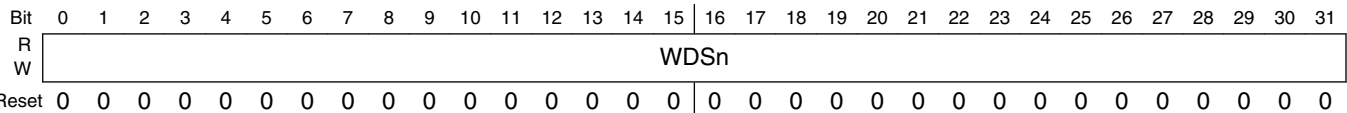
eTPU\_GCR\_WDSR\_ENG1 field descriptions

| Field        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>WDSn | Channel n Watchdog Status. For details about Watchdog mechanism, see <a href="#">Watchdog</a> .<br><br><b>NOTE:</b> The WDSn bits set in case of a watchdog timeout independently of <a href="#">WMO</a> .<br>When read:<br>0.....Watchdog did not expire during execution of a channel x thread since reset or last WDSn clear.<br>1.....Watchdog expired during execution of a channel n thread.<br>When written:<br>0.....Keep watchdog status bit unaltered.<br>1.....Clear watchdog status bit. |

### 35.12.14 eTPU Watchdog Status Register for Engine 2 (eTPU\_GCR\_WDSR\_ENG2)

This register indicates the watchdog influence on each of the engine channels.

Address: 0h base + 264h offset = 264h



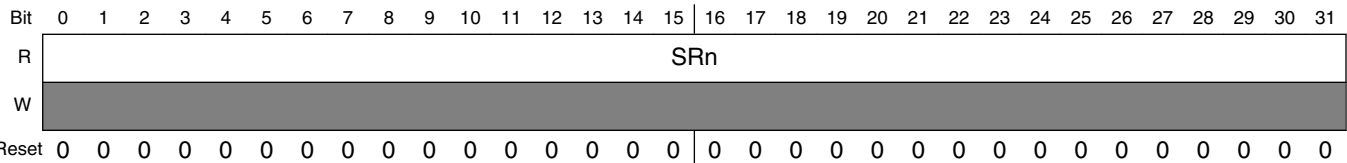
eTPU\_GCR\_WDSR\_ENG2 field descriptions

| Field        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>WDSn | <p>Channel n Watchdog Status. For details about Watchdog mechanism, see <a href="#">Watchdog</a>.</p> <p><b>NOTE:</b> The WDSn bits set in case of a watchdog timeout independently of <a href="#">WMO</a>.</p> <p>When read:</p> <p>0.....Watchdog did not expire during execution of a channel x thread since reset or last WDSn clear.</p> <p>1.....Watchdog expired during execution of a channel n thread.</p> <p>When written:</p> <p>0.....Keep watchdog status bit unaltered.</p> <p>1.....Clear watchdog status bit.</p> |

### 35.12.15 eTPU Channel Pending Service Status Register for Engine 1 (eTPU\_GCR\_CPSSR\_ENG1)

This register holds the status of the pending Channel Service Requests (see [Functions and threads](#)).

Address: 0h base + 280h offset = 280h



**eTPU\_GCR\_CPSSR\_ENG1 field descriptions**

| Field       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>SRn | <p>Pending Service Request n.</p> <p>Indicates a pending Service Request for channel n.</p> <p>Pending SR status is a logic OR of all service requests pending: if only HSR is active, SRn clears only at the end of the thread. SRn clear due to the other request sources is microcode dependent.</p> <p><b>NOTE:</b> The pending service status bit for a channel is 1 when a Service Request is pending, even if the Channel is disabled (CPRn = 0).</p> <p><b>NOTE:</b> There can be a delay of one clock between writing HSR greater than 0 in <a href="#">eTPU Channel n Host Service Request Register for Engine 1 (eTPU_CCCR_CnHSRR_ENG1)</a> of a channel and its respective bit being asserted in eTPU_GCR_CPSSR_ENG1.</p> <p>0 No Service Request pending for channel n.</p> <p>1 Pending Service Request for channel n.</p> |

**35.12.16 eTPU Channel Pending Service Status Register for Engine 2 (eTPU\_GCR\_CPSSR\_ENG2)**

This register is a read-only register that holds the status of the pending Channel Service Requests (see [Functions and threads](#)).

Address: 0h base + 284h offset = 284h

|       |     |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-----|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | SRn |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |     |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**eTPU\_GCR\_CPSSR\_ENG2 field descriptions**

| Field       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>SRn | <p>Pending Service Request n.</p> <p>Indicates a pending Service Request for channel n.</p> <p>Pending SR status is a logic OR of all service requests pending: if only HSR is active, SRn clears only at the end of the thread. SRn clear due to the other request sources is microcode dependent.</p> <p><b>NOTE:</b> The pending service status bit for a channel is 1 when a Service Request is pending, even if the Channel is disabled (CPRn = 0).</p> <p><b>NOTE:</b> There can be a delay of one clock between writing HSR greater than 0 in <a href="#">eTPU Channel n Host Service Request Register for Engine 2 (eTPU_CCCR_CnHSRR_ENG2)</a> of a channel and its respective bit being asserted in eTPU_GCR_CPSSR_ENG2.</p> <p>0 No Service Request pending for channel n.</p> <p>1 Pending Service Request for channel n.</p> |

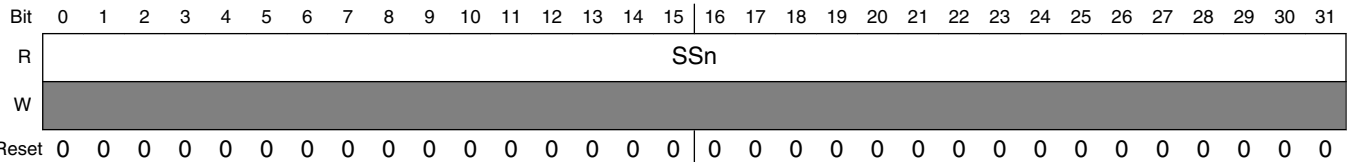
35.12.17 eTPU Channel Service Status Register for Engine 1  
(eTPU\_GCR\_CSSR\_ENG1)

This register holds the current channel service status on whether it is being serviced or not (see [Functions and threads](#)). Only one bit may be asserted in this register at a given time. When no channel is being serviced, the register read value is 0x0000\_0000. This register is read-only and can be read during normal eTPU operation for monitoring the scheduler activity.

NOTE

Channel Service Status does not always reflect decoding of the CHAN register (see [CHAN Register](#)), since the latter can be changed by the service thread microcode.

Address: 0h base + 290h offset = 290h



eTPU\_GCR\_CSSR\_ENG1 field descriptions

| Field       | Description                                                                                                                                                                                                                                                                                                                                  |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>SSn | <p>Service Status n.</p> <p>Indicates that channel n is currently being serviced. It is updated at the 1st microcycle of a Time Slot Transition (see <a href="#">Time Slot Transition</a>), or when the microengine ends the thread.</p> <p>0 Channel n is not currently being serviced.</p> <p>1 channel n is currently being serviced.</p> |

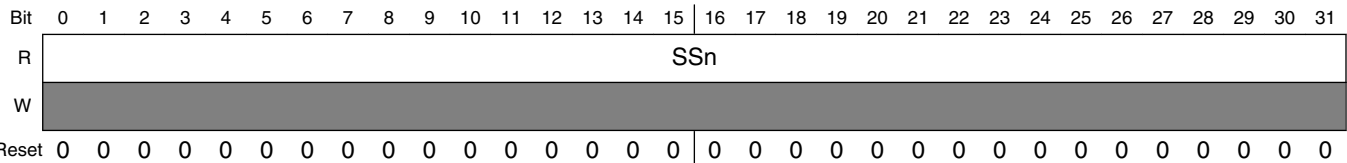
35.12.18 eTPU Channel Service Status Register for Engine 2  
(eTPU\_GCR\_CSSR\_ENG2)

This register holds the current channel service status on whether it is being serviced or not (see [Functions and threads](#)). Only one bit may be asserted in this register at a given time. When no channel is being serviced, the register read value is 0x0000\_0000. This read-only register can be read during normal eTPU operation for monitoring the scheduler activity.

NOTE

Channel Service Status does not always reflect decoding of the CHAN register (see [CHAN Register](#)), since the latter can be changed by the service thread microcode.

Address: 0h base + 294h offset = 294h



eTPU\_GCR\_CSSR\_ENG2 field descriptions

| Field       | Description                                                                                                                                                                                                                                                                                                                                  |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>SSn | <p>Service Status n.</p> <p>Indicates that channel n is currently being serviced. It is updated at the 1st microcycle of a Time Slot Transition (see <a href="#">Time Slot Transition</a>), or when the microengine ends the thread.</p> <p>0 Channel n is not currently being serviced.</p> <p>1 channel n is currently being serviced.</p> |

35.13 Channel Configuration Control memory map and registers

Each channel has a group of 3 registers used to control, configure and check status of that channel as shown in the following table. This organization eases individual channel management.

Table 35-121. Channel registers structure

| Channel offset | Register name                                                      |
|----------------|--------------------------------------------------------------------|
| 0x00           | eTPU_CCCR_CnCR_ENGm - eTPU Channel Configuration Register          |
| 0x04           | eTPU_CCCR_CnSCR_ENGm - eTPU Channel Status/Control Register        |
| 0x08           | eTPU_CCCR_CnHSRR_ENGm - eTPU Channel Host Service Request Register |
| 0x0C           | Reserved                                                           |

One contiguous area is used to map all channel registers of each eTPU engine as shown in the following table.

**Table 35-122. Channel registers mMap**

| Offset | Registers structure                   |
|--------|---------------------------------------|
| 0x400  | eTPU 1 Channel 0 Registers Structure  |
| 0x410  | eTPU 1 Channel 1 Registers Structure  |
| 0x420  | eTPU 1 Channel 2 Registers Structure  |
| 0x430  | ----                                  |
| 0x5E0  | eTPU 1 Channel 30 Registers Structure |
| 0x5F0  | eTPU 1 Channel 31 Registers Structure |
| 0x600  | RESERVED                              |
| 0x800  | eTPU 2 Channel 0 Registers Structure  |
| 0x810  | eTPU 2 Channel 1 Registers Structure  |
| 0x820  | ----                                  |
| 0x9E0  | eTPU 2 Channel 30 Registers Structure |
| 0x9F0  | eTPU 2 Channel 31 Registers Structure |
| 0xA00  | RESERVED                              |

There are 64 structures defined, one for each available channel in the eTPU System (32 for each Engine). The base address for the structure presented can be calculated by using the following equation:

$\text{Channel\_Register\_Base} = \text{eTPU\_Engine\_Channel\_Base} + (\text{channel\_number} * 0x10)$ ,  
where:

- eTPU\_Engine\_Channel\_Base = eTPU\_Base + 0x400 for Engine 1.
- eTPU\_Engine\_Channel\_Base = eTPU\_Base + 0x800 for Engine 2.

#### eTPU\_CCCR memory map

| Address offset (hex) | Register name                                                                     | Width (in bits) | Access | Reset value                 | Section/page                 |
|----------------------|-----------------------------------------------------------------------------------|-----------------|--------|-----------------------------|------------------------------|
| 400                  | eTPU Channel n Configuration Register for Engine 1 (eTPU_CCCR_C0CR_ENG1)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.1/1387</a> |
| 404                  | eTPU Channel n Status Control Register for Engine 1 (eTPU_CCCR_C0SCR_ENG1)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.2/1390</a> |
| 408                  | eTPU Channel n Host Service Request Register for Engine 1 (eTPU_CCCR_C0HSRR_ENG1) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.3/1392</a> |
| 410                  | eTPU Channel n Configuration Register for Engine 1 (eTPU_CCCR_C1CR_ENG1)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.1/1387</a> |
| 414                  | eTPU Channel n Status Control Register for Engine 1 (eTPU_CCCR_C1SCR_ENG1)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.2/1390</a> |
| 418                  | eTPU Channel n Host Service Request Register for Engine 1 (eTPU_CCCR_C1HSRR_ENG1) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.3/1392</a> |

*Table continues on the next page...*

**eTPU\_CCCR memory map (continued)**

| Address offset (hex) | Register name                                                                     | Width (in bits) | Access | Reset value                 | Section/ page                |
|----------------------|-----------------------------------------------------------------------------------|-----------------|--------|-----------------------------|------------------------------|
| 420                  | eTPU Channel n Configuration Register for Engine 1 (eTPU_CCCR_C2CR_ENG1)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.1/1387</a> |
| 424                  | eTPU Channel n Status Control Register for Engine 1 (eTPU_CCCR_C2SCR_ENG1)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.2/1390</a> |
| 428                  | eTPU Channel n Host Service Request Register for Engine 1 (eTPU_CCCR_C2HSRR_ENG1) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.3/1392</a> |
| 430                  | eTPU Channel n Configuration Register for Engine 1 (eTPU_CCCR_C3CR_ENG1)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.1/1387</a> |
| 434                  | eTPU Channel n Status Control Register for Engine 1 (eTPU_CCCR_C3SCR_ENG1)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.2/1390</a> |
| 438                  | eTPU Channel n Host Service Request Register for Engine 1 (eTPU_CCCR_C3HSRR_ENG1) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.3/1392</a> |
| 440                  | eTPU Channel n Configuration Register for Engine 1 (eTPU_CCCR_C4CR_ENG1)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.1/1387</a> |
| 444                  | eTPU Channel n Status Control Register for Engine 1 (eTPU_CCCR_C4SCR_ENG1)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.2/1390</a> |
| 448                  | eTPU Channel n Host Service Request Register for Engine 1 (eTPU_CCCR_C4HSRR_ENG1) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.3/1392</a> |
| 450                  | eTPU Channel n Configuration Register for Engine 1 (eTPU_CCCR_C5CR_ENG1)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.1/1387</a> |
| 454                  | eTPU Channel n Status Control Register for Engine 1 (eTPU_CCCR_C5SCR_ENG1)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.2/1390</a> |
| 458                  | eTPU Channel n Host Service Request Register for Engine 1 (eTPU_CCCR_C5HSRR_ENG1) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.3/1392</a> |
| 460                  | eTPU Channel n Configuration Register for Engine 1 (eTPU_CCCR_C6CR_ENG1)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.1/1387</a> |
| 464                  | eTPU Channel n Status Control Register for Engine 1 (eTPU_CCCR_C6SCR_ENG1)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.2/1390</a> |
| 468                  | eTPU Channel n Host Service Request Register for Engine 1 (eTPU_CCCR_C6HSRR_ENG1) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.3/1392</a> |
| 470                  | eTPU Channel n Configuration Register for Engine 1 (eTPU_CCCR_C7CR_ENG1)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.1/1387</a> |
| 474                  | eTPU Channel n Status Control Register for Engine 1 (eTPU_CCCR_C7SCR_ENG1)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.2/1390</a> |
| 478                  | eTPU Channel n Host Service Request Register for Engine 1 (eTPU_CCCR_C7HSRR_ENG1) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.3/1392</a> |
| 480                  | eTPU Channel n Configuration Register for Engine 1 (eTPU_CCCR_C8CR_ENG1)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.1/1387</a> |
| 484                  | eTPU Channel n Status Control Register for Engine 1 (eTPU_CCCR_C8SCR_ENG1)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.2/1390</a> |
| 488                  | eTPU Channel n Host Service Request Register for Engine 1 (eTPU_CCCR_C8HSRR_ENG1) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.3/1392</a> |
| 490                  | eTPU Channel n Configuration Register for Engine 1 (eTPU_CCCR_C9CR_ENG1)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.1/1387</a> |

*Table continues on the next page...*

**eTPU\_CCCR memory map (continued)**

| Address offset (hex) | Register name                                                                      | Width (in bits) | Access | Reset value                 | Section/ page                |
|----------------------|------------------------------------------------------------------------------------|-----------------|--------|-----------------------------|------------------------------|
| 494                  | eTPU Channel n Status Control Register for Engine 1 (eTPU_CCCR_C9SCR_ENG1)         | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.2/1390</a> |
| 498                  | eTPU Channel n Host Service Request Register for Engine 1 (eTPU_CCCR_C9HSRR_ENG1)  | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.3/1392</a> |
| 4A0                  | eTPU Channel n Configuration Register for Engine 1 (eTPU_CCCR_C10CR_ENG1)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.1/1387</a> |
| 4A4                  | eTPU Channel n Status Control Register for Engine 1 (eTPU_CCCR_C10SCR_ENG1)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.2/1390</a> |
| 4A8                  | eTPU Channel n Host Service Request Register for Engine 1 (eTPU_CCCR_C10HSRR_ENG1) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.3/1392</a> |
| 4B0                  | eTPU Channel n Configuration Register for Engine 1 (eTPU_CCCR_C11CR_ENG1)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.1/1387</a> |
| 4B4                  | eTPU Channel n Status Control Register for Engine 1 (eTPU_CCCR_C11SCR_ENG1)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.2/1390</a> |
| 4B8                  | eTPU Channel n Host Service Request Register for Engine 1 (eTPU_CCCR_C11HSRR_ENG1) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.3/1392</a> |
| 4C0                  | eTPU Channel n Configuration Register for Engine 1 (eTPU_CCCR_C12CR_ENG1)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.1/1387</a> |
| 4C4                  | eTPU Channel n Status Control Register for Engine 1 (eTPU_CCCR_C12SCR_ENG1)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.2/1390</a> |
| 4C8                  | eTPU Channel n Host Service Request Register for Engine 1 (eTPU_CCCR_C12HSRR_ENG1) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.3/1392</a> |
| 4D0                  | eTPU Channel n Configuration Register for Engine 1 (eTPU_CCCR_C13CR_ENG1)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.1/1387</a> |
| 4D4                  | eTPU Channel n Status Control Register for Engine 1 (eTPU_CCCR_C13SCR_ENG1)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.2/1390</a> |
| 4D8                  | eTPU Channel n Host Service Request Register for Engine 1 (eTPU_CCCR_C13HSRR_ENG1) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.3/1392</a> |
| 4E0                  | eTPU Channel n Configuration Register for Engine 1 (eTPU_CCCR_C14CR_ENG1)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.1/1387</a> |
| 4E4                  | eTPU Channel n Status Control Register for Engine 1 (eTPU_CCCR_C14SCR_ENG1)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.2/1390</a> |
| 4E8                  | eTPU Channel n Host Service Request Register for Engine 1 (eTPU_CCCR_C14HSRR_ENG1) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.3/1392</a> |
| 4F0                  | eTPU Channel n Configuration Register for Engine 1 (eTPU_CCCR_C15CR_ENG1)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.1/1387</a> |
| 4F4                  | eTPU Channel n Status Control Register for Engine 1 (eTPU_CCCR_C15SCR_ENG1)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.2/1390</a> |
| 4F8                  | eTPU Channel n Host Service Request Register for Engine 1 (eTPU_CCCR_C15HSRR_ENG1) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.3/1392</a> |
| 500                  | eTPU Channel n Configuration Register for Engine 1 (eTPU_CCCR_C16CR_ENG1)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.1/1387</a> |
| 504                  | eTPU Channel n Status Control Register for Engine 1 (eTPU_CCCR_C16SCR_ENG1)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.2/1390</a> |

*Table continues on the next page...*



**eTPU\_CCCR memory map (continued)**

| Address offset (hex) | Register name                                                                      | Width (in bits) | Access | Reset value                 | Section/ page                |
|----------------------|------------------------------------------------------------------------------------|-----------------|--------|-----------------------------|------------------------------|
| 508                  | eTPU Channel n Host Service Request Register for Engine 1 (eTPU_CCCR_C16HSRR_ENG1) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.3/1392</a> |
| 510                  | eTPU Channel n Configuration Register for Engine 1 (eTPU_CCCR_C17CR_ENG1)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.1/1387</a> |
| 514                  | eTPU Channel n Status Control Register for Engine 1 (eTPU_CCCR_C17SCR_ENG1)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.2/1390</a> |
| 518                  | eTPU Channel n Host Service Request Register for Engine 1 (eTPU_CCCR_C17HSRR_ENG1) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.3/1392</a> |
| 520                  | eTPU Channel n Configuration Register for Engine 1 (eTPU_CCCR_C18CR_ENG1)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.1/1387</a> |
| 524                  | eTPU Channel n Status Control Register for Engine 1 (eTPU_CCCR_C18SCR_ENG1)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.2/1390</a> |
| 528                  | eTPU Channel n Host Service Request Register for Engine 1 (eTPU_CCCR_C18HSRR_ENG1) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.3/1392</a> |
| 530                  | eTPU Channel n Configuration Register for Engine 1 (eTPU_CCCR_C19CR_ENG1)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.1/1387</a> |
| 534                  | eTPU Channel n Status Control Register for Engine 1 (eTPU_CCCR_C19SCR_ENG1)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.2/1390</a> |
| 538                  | eTPU Channel n Host Service Request Register for Engine 1 (eTPU_CCCR_C19HSRR_ENG1) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.3/1392</a> |
| 540                  | eTPU Channel n Configuration Register for Engine 1 (eTPU_CCCR_C20CR_ENG1)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.1/1387</a> |
| 544                  | eTPU Channel n Status Control Register for Engine 1 (eTPU_CCCR_C20SCR_ENG1)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.2/1390</a> |
| 548                  | eTPU Channel n Host Service Request Register for Engine 1 (eTPU_CCCR_C20HSRR_ENG1) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.3/1392</a> |
| 550                  | eTPU Channel n Configuration Register for Engine 1 (eTPU_CCCR_C21CR_ENG1)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.1/1387</a> |
| 554                  | eTPU Channel n Status Control Register for Engine 1 (eTPU_CCCR_C21SCR_ENG1)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.2/1390</a> |
| 558                  | eTPU Channel n Host Service Request Register for Engine 1 (eTPU_CCCR_C21HSRR_ENG1) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.3/1392</a> |
| 560                  | eTPU Channel n Configuration Register for Engine 1 (eTPU_CCCR_C22CR_ENG1)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.1/1387</a> |
| 564                  | eTPU Channel n Status Control Register for Engine 1 (eTPU_CCCR_C22SCR_ENG1)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.2/1390</a> |
| 568                  | eTPU Channel n Host Service Request Register for Engine 1 (eTPU_CCCR_C22HSRR_ENG1) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.3/1392</a> |
| 570                  | eTPU Channel n Configuration Register for Engine 1 (eTPU_CCCR_C23CR_ENG1)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.1/1387</a> |
| 574                  | eTPU Channel n Status Control Register for Engine 1 (eTPU_CCCR_C23SCR_ENG1)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.2/1390</a> |
| 578                  | eTPU Channel n Host Service Request Register for Engine 1 (eTPU_CCCR_C23HSRR_ENG1) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.3/1392</a> |

*Table continues on the next page...*

**eTPU\_CCCR memory map (continued)**

| Address offset (hex) | Register name                                                                      | Width (in bits) | Access | Reset value                 | Section/page                 |
|----------------------|------------------------------------------------------------------------------------|-----------------|--------|-----------------------------|------------------------------|
| 580                  | eTPU Channel n Configuration Register for Engine 1 (eTPU_CCCR_C24CR_ENG1)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.1/1387</a> |
| 584                  | eTPU Channel n Status Control Register for Engine 1 (eTPU_CCCR_C24SCR_ENG1)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.2/1390</a> |
| 588                  | eTPU Channel n Host Service Request Register for Engine 1 (eTPU_CCCR_C24HSRR_ENG1) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.3/1392</a> |
| 590                  | eTPU Channel n Configuration Register for Engine 1 (eTPU_CCCR_C25CR_ENG1)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.1/1387</a> |
| 594                  | eTPU Channel n Status Control Register for Engine 1 (eTPU_CCCR_C25SCR_ENG1)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.2/1390</a> |
| 598                  | eTPU Channel n Host Service Request Register for Engine 1 (eTPU_CCCR_C25HSRR_ENG1) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.3/1392</a> |
| 5A0                  | eTPU Channel n Configuration Register for Engine 1 (eTPU_CCCR_C26CR_ENG1)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.1/1387</a> |
| 5A4                  | eTPU Channel n Status Control Register for Engine 1 (eTPU_CCCR_C26SCR_ENG1)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.2/1390</a> |
| 5A8                  | eTPU Channel n Host Service Request Register for Engine 1 (eTPU_CCCR_C26HSRR_ENG1) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.3/1392</a> |
| 5B0                  | eTPU Channel n Configuration Register for Engine 1 (eTPU_CCCR_C27CR_ENG1)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.1/1387</a> |
| 5B4                  | eTPU Channel n Status Control Register for Engine 1 (eTPU_CCCR_C27SCR_ENG1)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.2/1390</a> |
| 5B8                  | eTPU Channel n Host Service Request Register for Engine 1 (eTPU_CCCR_C27HSRR_ENG1) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.3/1392</a> |
| 5C0                  | eTPU Channel n Configuration Register for Engine 1 (eTPU_CCCR_C28CR_ENG1)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.1/1387</a> |
| 5C4                  | eTPU Channel n Status Control Register for Engine 1 (eTPU_CCCR_C28SCR_ENG1)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.2/1390</a> |
| 5C8                  | eTPU Channel n Host Service Request Register for Engine 1 (eTPU_CCCR_C28HSRR_ENG1) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.3/1392</a> |
| 5D0                  | eTPU Channel n Configuration Register for Engine 1 (eTPU_CCCR_C29CR_ENG1)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.1/1387</a> |
| 5D4                  | eTPU Channel n Status Control Register for Engine 1 (eTPU_CCCR_C29SCR_ENG1)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.2/1390</a> |
| 5D8                  | eTPU Channel n Host Service Request Register for Engine 1 (eTPU_CCCR_C29HSRR_ENG1) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.3/1392</a> |
| 5E0                  | eTPU Channel n Configuration Register for Engine 1 (eTPU_CCCR_C30CR_ENG1)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.1/1387</a> |
| 5E4                  | eTPU Channel n Status Control Register for Engine 1 (eTPU_CCCR_C30SCR_ENG1)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.2/1390</a> |
| 5E8                  | eTPU Channel n Host Service Request Register for Engine 1 (eTPU_CCCR_C30HSRR_ENG1) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.3/1392</a> |
| 5F0                  | eTPU Channel n Configuration Register for Engine 1 (eTPU_CCCR_C31CR_ENG1)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.1/1387</a> |

*Table continues on the next page...*

**eTPU\_CCCR memory map (continued)**

| Address offset (hex) | Register name                                                                      | Width (in bits) | Access | Reset value                 | Section/page                 |
|----------------------|------------------------------------------------------------------------------------|-----------------|--------|-----------------------------|------------------------------|
| 5F4                  | eTPU Channel n Status Control Register for Engine 1 (eTPU_CCCR_C31SCR_ENG1)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.2/1390</a> |
| 5F8                  | eTPU Channel n Host Service Request Register for Engine 1 (eTPU_CCCR_C31HSRR_ENG1) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.3/1392</a> |
| 800                  | eTPU Channel n Configuration Register for Engine 2 (eTPU_CCCR_C0CR_ENG2)           | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.4/1393</a> |
| 804                  | eTPU Channel n Status Control Register for Engine 2 (eTPU_CCCR_C0SCR_ENG2)         | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.5/1396</a> |
| 808                  | eTPU Channel n Host Service Request Register for Engine 2 (eTPU_CCCR_C0HSRR_ENG2)  | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.6/1398</a> |
| 810                  | eTPU Channel n Configuration Register for Engine 2 (eTPU_CCCR_C1CR_ENG2)           | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.4/1393</a> |
| 814                  | eTPU Channel n Status Control Register for Engine 2 (eTPU_CCCR_C1SCR_ENG2)         | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.5/1396</a> |
| 818                  | eTPU Channel n Host Service Request Register for Engine 2 (eTPU_CCCR_C1HSRR_ENG2)  | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.6/1398</a> |
| 820                  | eTPU Channel n Configuration Register for Engine 2 (eTPU_CCCR_C2CR_ENG2)           | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.4/1393</a> |
| 824                  | eTPU Channel n Status Control Register for Engine 2 (eTPU_CCCR_C2SCR_ENG2)         | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.5/1396</a> |
| 828                  | eTPU Channel n Host Service Request Register for Engine 2 (eTPU_CCCR_C2HSRR_ENG2)  | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.6/1398</a> |
| 830                  | eTPU Channel n Configuration Register for Engine 2 (eTPU_CCCR_C3CR_ENG2)           | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.4/1393</a> |
| 834                  | eTPU Channel n Status Control Register for Engine 2 (eTPU_CCCR_C3SCR_ENG2)         | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.5/1396</a> |
| 838                  | eTPU Channel n Host Service Request Register for Engine 2 (eTPU_CCCR_C3HSRR_ENG2)  | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.6/1398</a> |
| 840                  | eTPU Channel n Configuration Register for Engine 2 (eTPU_CCCR_C4CR_ENG2)           | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.4/1393</a> |
| 844                  | eTPU Channel n Status Control Register for Engine 2 (eTPU_CCCR_C4SCR_ENG2)         | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.5/1396</a> |
| 848                  | eTPU Channel n Host Service Request Register for Engine 2 (eTPU_CCCR_C4HSRR_ENG2)  | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.6/1398</a> |
| 850                  | eTPU Channel n Configuration Register for Engine 2 (eTPU_CCCR_C5CR_ENG2)           | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.4/1393</a> |
| 854                  | eTPU Channel n Status Control Register for Engine 2 (eTPU_CCCR_C5SCR_ENG2)         | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.5/1396</a> |
| 858                  | eTPU Channel n Host Service Request Register for Engine 2 (eTPU_CCCR_C5HSRR_ENG2)  | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.6/1398</a> |
| 860                  | eTPU Channel n Configuration Register for Engine 2 (eTPU_CCCR_C6CR_ENG2)           | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.4/1393</a> |
| 864                  | eTPU Channel n Status Control Register for Engine 2 (eTPU_CCCR_C6SCR_ENG2)         | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.5/1396</a> |

Table continues on the next page...

**eTPU\_CCCR memory map (continued)**

| Address offset (hex) | Register name                                                                      | Width (in bits) | Access | Reset value                 | Section/ page                |
|----------------------|------------------------------------------------------------------------------------|-----------------|--------|-----------------------------|------------------------------|
| 868                  | eTPU Channel n Host Service Request Register for Engine 2 (eTPU_CCCR_C6HSRR_ENG2)  | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.6/1398</a> |
| 870                  | eTPU Channel n Configuration Register for Engine 2 (eTPU_CCCR_C7CR_ENG2)           | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.4/1393</a> |
| 874                  | eTPU Channel n Status Control Register for Engine 2 (eTPU_CCCR_C7SCR_ENG2)         | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.5/1396</a> |
| 878                  | eTPU Channel n Host Service Request Register for Engine 2 (eTPU_CCCR_C7HSRR_ENG2)  | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.6/1398</a> |
| 880                  | eTPU Channel n Configuration Register for Engine 2 (eTPU_CCCR_C8CR_ENG2)           | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.4/1393</a> |
| 884                  | eTPU Channel n Status Control Register for Engine 2 (eTPU_CCCR_C8SCR_ENG2)         | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.5/1396</a> |
| 888                  | eTPU Channel n Host Service Request Register for Engine 2 (eTPU_CCCR_C8HSRR_ENG2)  | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.6/1398</a> |
| 890                  | eTPU Channel n Configuration Register for Engine 2 (eTPU_CCCR_C9CR_ENG2)           | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.4/1393</a> |
| 894                  | eTPU Channel n Status Control Register for Engine 2 (eTPU_CCCR_C9SCR_ENG2)         | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.5/1396</a> |
| 898                  | eTPU Channel n Host Service Request Register for Engine 2 (eTPU_CCCR_C9HSRR_ENG2)  | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.6/1398</a> |
| 8A0                  | eTPU Channel n Configuration Register for Engine 2 (eTPU_CCCR_C10CR_ENG2)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.4/1393</a> |
| 8A4                  | eTPU Channel n Status Control Register for Engine 2 (eTPU_CCCR_C10SCR_ENG2)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.5/1396</a> |
| 8A8                  | eTPU Channel n Host Service Request Register for Engine 2 (eTPU_CCCR_C10HSRR_ENG2) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.6/1398</a> |
| 8B0                  | eTPU Channel n Configuration Register for Engine 2 (eTPU_CCCR_C11CR_ENG2)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.4/1393</a> |
| 8B4                  | eTPU Channel n Status Control Register for Engine 2 (eTPU_CCCR_C11SCR_ENG2)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.5/1396</a> |
| 8B8                  | eTPU Channel n Host Service Request Register for Engine 2 (eTPU_CCCR_C11HSRR_ENG2) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.6/1398</a> |
| 8C0                  | eTPU Channel n Configuration Register for Engine 2 (eTPU_CCCR_C12CR_ENG2)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.4/1393</a> |
| 8C4                  | eTPU Channel n Status Control Register for Engine 2 (eTPU_CCCR_C12SCR_ENG2)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.5/1396</a> |
| 8C8                  | eTPU Channel n Host Service Request Register for Engine 2 (eTPU_CCCR_C12HSRR_ENG2) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.6/1398</a> |
| 8D0                  | eTPU Channel n Configuration Register for Engine 2 (eTPU_CCCR_C13CR_ENG2)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.4/1393</a> |
| 8D4                  | eTPU Channel n Status Control Register for Engine 2 (eTPU_CCCR_C13SCR_ENG2)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.5/1396</a> |
| 8D8                  | eTPU Channel n Host Service Request Register for Engine 2 (eTPU_CCCR_C13HSRR_ENG2) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.6/1398</a> |

*Table continues on the next page...*

**eTPU\_CCCR memory map (continued)**

| Address offset (hex) | Register name                                                                      | Width (in bits) | Access | Reset value                 | Section/ page                |
|----------------------|------------------------------------------------------------------------------------|-----------------|--------|-----------------------------|------------------------------|
| 8E0                  | eTPU Channel n Configuration Register for Engine 2 (eTPU_CCCR_C14CR_ENG2)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.4/1393</a> |
| 8E4                  | eTPU Channel n Status Control Register for Engine 2 (eTPU_CCCR_C14SCR_ENG2)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.5/1396</a> |
| 8E8                  | eTPU Channel n Host Service Request Register for Engine 2 (eTPU_CCCR_C14HSRR_ENG2) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.6/1398</a> |
| 8F0                  | eTPU Channel n Configuration Register for Engine 2 (eTPU_CCCR_C15CR_ENG2)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.4/1393</a> |
| 8F4                  | eTPU Channel n Status Control Register for Engine 2 (eTPU_CCCR_C15SCR_ENG2)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.5/1396</a> |
| 8F8                  | eTPU Channel n Host Service Request Register for Engine 2 (eTPU_CCCR_C15HSRR_ENG2) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.6/1398</a> |
| 900                  | eTPU Channel n Configuration Register for Engine 2 (eTPU_CCCR_C16CR_ENG2)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.4/1393</a> |
| 904                  | eTPU Channel n Status Control Register for Engine 2 (eTPU_CCCR_C16SCR_ENG2)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.5/1396</a> |
| 908                  | eTPU Channel n Host Service Request Register for Engine 2 (eTPU_CCCR_C16HSRR_ENG2) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.6/1398</a> |
| 910                  | eTPU Channel n Configuration Register for Engine 2 (eTPU_CCCR_C17CR_ENG2)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.4/1393</a> |
| 914                  | eTPU Channel n Status Control Register for Engine 2 (eTPU_CCCR_C17SCR_ENG2)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.5/1396</a> |
| 918                  | eTPU Channel n Host Service Request Register for Engine 2 (eTPU_CCCR_C17HSRR_ENG2) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.6/1398</a> |
| 920                  | eTPU Channel n Configuration Register for Engine 2 (eTPU_CCCR_C18CR_ENG2)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.4/1393</a> |
| 924                  | eTPU Channel n Status Control Register for Engine 2 (eTPU_CCCR_C18SCR_ENG2)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.5/1396</a> |
| 928                  | eTPU Channel n Host Service Request Register for Engine 2 (eTPU_CCCR_C18HSRR_ENG2) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.6/1398</a> |
| 930                  | eTPU Channel n Configuration Register for Engine 2 (eTPU_CCCR_C19CR_ENG2)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.4/1393</a> |
| 934                  | eTPU Channel n Status Control Register for Engine 2 (eTPU_CCCR_C19SCR_ENG2)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.5/1396</a> |
| 938                  | eTPU Channel n Host Service Request Register for Engine 2 (eTPU_CCCR_C19HSRR_ENG2) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.6/1398</a> |
| 940                  | eTPU Channel n Configuration Register for Engine 2 (eTPU_CCCR_C20CR_ENG2)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.4/1393</a> |
| 944                  | eTPU Channel n Status Control Register for Engine 2 (eTPU_CCCR_C20SCR_ENG2)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.5/1396</a> |
| 948                  | eTPU Channel n Host Service Request Register for Engine 2 (eTPU_CCCR_C20HSRR_ENG2) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.6/1398</a> |
| 950                  | eTPU Channel n Configuration Register for Engine 2 (eTPU_CCCR_C21CR_ENG2)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.4/1393</a> |

*Table continues on the next page...*

**eTPU\_CCCR memory map (continued)**

| Address offset (hex) | Register name                                                                      | Width (in bits) | Access | Reset value                 | Section/ page                |
|----------------------|------------------------------------------------------------------------------------|-----------------|--------|-----------------------------|------------------------------|
| 954                  | eTPU Channel n Status Control Register for Engine 2 (eTPU_CCCR_C21SCR_ENG2)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.5/1396</a> |
| 958                  | eTPU Channel n Host Service Request Register for Engine 2 (eTPU_CCCR_C21HSRR_ENG2) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.6/1398</a> |
| 960                  | eTPU Channel n Configuration Register for Engine 2 (eTPU_CCCR_C22CR_ENG2)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.4/1393</a> |
| 964                  | eTPU Channel n Status Control Register for Engine 2 (eTPU_CCCR_C22SCR_ENG2)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.5/1396</a> |
| 968                  | eTPU Channel n Host Service Request Register for Engine 2 (eTPU_CCCR_C22HSRR_ENG2) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.6/1398</a> |
| 970                  | eTPU Channel n Configuration Register for Engine 2 (eTPU_CCCR_C23CR_ENG2)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.4/1393</a> |
| 974                  | eTPU Channel n Status Control Register for Engine 2 (eTPU_CCCR_C23SCR_ENG2)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.5/1396</a> |
| 978                  | eTPU Channel n Host Service Request Register for Engine 2 (eTPU_CCCR_C23HSRR_ENG2) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.6/1398</a> |
| 980                  | eTPU Channel n Configuration Register for Engine 2 (eTPU_CCCR_C24CR_ENG2)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.4/1393</a> |
| 984                  | eTPU Channel n Status Control Register for Engine 2 (eTPU_CCCR_C24SCR_ENG2)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.5/1396</a> |
| 988                  | eTPU Channel n Host Service Request Register for Engine 2 (eTPU_CCCR_C24HSRR_ENG2) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.6/1398</a> |
| 990                  | eTPU Channel n Configuration Register for Engine 2 (eTPU_CCCR_C25CR_ENG2)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.4/1393</a> |
| 994                  | eTPU Channel n Status Control Register for Engine 2 (eTPU_CCCR_C25SCR_ENG2)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.5/1396</a> |
| 998                  | eTPU Channel n Host Service Request Register for Engine 2 (eTPU_CCCR_C25HSRR_ENG2) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.6/1398</a> |
| 9A0                  | eTPU Channel n Configuration Register for Engine 2 (eTPU_CCCR_C26CR_ENG2)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.4/1393</a> |
| 9A4                  | eTPU Channel n Status Control Register for Engine 2 (eTPU_CCCR_C26SCR_ENG2)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.5/1396</a> |
| 9A8                  | eTPU Channel n Host Service Request Register for Engine 2 (eTPU_CCCR_C26HSRR_ENG2) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.6/1398</a> |
| 9B0                  | eTPU Channel n Configuration Register for Engine 2 (eTPU_CCCR_C27CR_ENG2)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.4/1393</a> |
| 9B4                  | eTPU Channel n Status Control Register for Engine 2 (eTPU_CCCR_C27SCR_ENG2)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.5/1396</a> |
| 9B8                  | eTPU Channel n Host Service Request Register for Engine 2 (eTPU_CCCR_C27HSRR_ENG2) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.6/1398</a> |
| 9C0                  | eTPU Channel n Configuration Register for Engine 2 (eTPU_CCCR_C28CR_ENG2)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.4/1393</a> |
| 9C4                  | eTPU Channel n Status Control Register for Engine 2 (eTPU_CCCR_C28SCR_ENG2)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.5/1396</a> |

*Table continues on the next page...*



**eTPU\_CCCR memory map (continued)**

| Address offset (hex) | Register name                                                                      | Width (in bits) | Access | Reset value                 | Section/ page                |
|----------------------|------------------------------------------------------------------------------------|-----------------|--------|-----------------------------|------------------------------|
| 9C8                  | eTPU Channel n Host Service Request Register for Engine 2 (eTPU_CCCR_C28HSRR_ENG2) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.6/1398</a> |
| 9D0                  | eTPU Channel n Configuration Register for Engine 2 (eTPU_CCCR_C29CR_ENG2)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.4/1393</a> |
| 9D4                  | eTPU Channel n Status Control Register for Engine 2 (eTPU_CCCR_C29SCR_ENG2)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.5/1396</a> |
| 9D8                  | eTPU Channel n Host Service Request Register for Engine 2 (eTPU_CCCR_C29HSRR_ENG2) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.6/1398</a> |
| 9E0                  | eTPU Channel n Configuration Register for Engine 2 (eTPU_CCCR_C30CR_ENG2)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.4/1393</a> |
| 9E4                  | eTPU Channel n Status Control Register for Engine 2 (eTPU_CCCR_C30SCR_ENG2)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.5/1396</a> |
| 9E8                  | eTPU Channel n Host Service Request Register for Engine 2 (eTPU_CCCR_C30HSRR_ENG2) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.6/1398</a> |
| 9F0                  | eTPU Channel n Configuration Register for Engine 2 (eTPU_CCCR_C31CR_ENG2)          | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.4/1393</a> |
| 9F4                  | eTPU Channel n Status Control Register for Engine 2 (eTPU_CCCR_C31SCR_ENG2)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">35.13.5/1396</a> |
| 9F8                  | eTPU Channel n Host Service Request Register for Engine 2 (eTPU_CCCR_C31HSRR_ENG2) | 32              | R/W    | 0000_0000h                  | <a href="#">35.13.6/1398</a> |

### 35.13.1 eTPU Channel n Configuration Register for Engine 1 (eTPU\_CCCR\_CnCR\_ENG1)

This register gathers configurations set individually per channel.

Address: 0h base + 400h offset + (16d × i), where i=0d to 31d

|       |      |      |     |       |       |      |      |    |    |    |    |    |    |    |    |    |
|-------|------|------|-----|-------|-------|------|------|----|----|----|----|----|----|----|----|----|
| Bit   | 0    | 1    | 2   | 3     | 4     | 5    | 6    | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
| R     |      |      |     |       |       |      |      |    |    | 0  |    |    |    |    |    |    |
| W     | CIE  | DTRE | CPR | T4DIS | SSMAE | ETPD | ETCS |    |    |    |    |    |    |    |    |    |
| Reset | 0    | 0    | 0   | 0     | 0     | 0    | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| Bit   | 16   | 17   | 18  | 19    | 20    | 21   | 22   | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |      |      |     |       |       |      |      |    |    |    |    |    |    |    |    |    |
| W     | ODIS | OPOL |     |       |       |      |      |    |    |    |    |    |    |    |    |    |
| Reset | 0    | 0    | 0   | 0     | 0     | 0    | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**eTPU\_CCCR\_CnCR\_ENG1 field descriptions**

| Field      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>CIE   | <p>Channel Interrupt Enable (see <a href="#">Channel Interrupt and Data Transfer Requests</a>).</p> <p>This bit is mirrored from <a href="#">eTPU Channel Interrupt Enable Register for Engine 1 (eTPU_GCR_CIER_ENG1)</a>.</p> <p>0 Disable interrupt for this channel.<br/>1 Enable interrupt for this channel.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 1<br>DTRE  | <p>Channel Data Transfer Request Enable (see <a href="#">Interrupts and data transfer requests</a>).</p> <p>This bit is mirrored from <a href="#">eTPU Channel Data Transfer Request Enable Register for Engine 1 (eTPU_GCR_CDTRER_ENG1)</a>.</p> <p>0 Disable data transfer request for this channel.<br/>1 Enable data transfer request for this channel.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 2–3<br>CPR | <p>Channel Priority.</p> <p>This field defines the priority level for the channel, used by the Hardware Scheduler (see <a href="#">Scheduler</a>).</p> <p>00 Disabled<br/>01 Low<br/>10 Middle<br/>11 High</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 4<br>T4DIS | <p>T4 channel action Disable.</p> <p>This bit forces the channel to work on T2 timing mode, regardless of the FCSS and CDFC fields in <a href="#">eTPU Engine 1 Configuration Register (eTPU_SCR_ECR_ENG1)</a>, or of the TCR1CS field in <a href="#">eTPU Engine 1 Time Base Configuration Register (eTPU_TBR_TBCR_ENG1)</a>. For more details, see <a href="#">Channel timing modes</a>.</p> <p>0 Channel works either in T2 or T2/T4 timing modes, depending on the TCR1CS field in <a href="#">eTPU Engine 1 Time Base Configuration Register (eTPU_TBR_TBCR_ENG1)</a> and the FCSS and CDFC fields in <a href="#">eTPU Engine 1 Configuration Register (eTPU_SCR_ECR_ENG1)</a>.<br/>1 Channel works in T2 timing mode, regardless of the TCR1CS field in <a href="#">eTPU Engine 1 Time Base Configuration Register (eTPU_TBR_TBCR_ENG1)</a> or the FCSS and CDFC fields in <a href="#">eTPU Engine 1 Configuration Register (eTPU_SCR_ECR_ENG1)</a>.</p> |
| 5<br>SSMAE | <p>Shared Subsystem Microengine Access Enable.</p> <p>This bit enables the microengine access to the Shared Subsystem Area (see <a href="#">Shared Subsystems support</a>). When this bit is negated, DIOB address values pointing to the Shared Subsystem Area actually access SDM mirrors, in legacy mode.</p> <p><b>NOTE:</b> SSMAE value must only be changed when the engine is either stopped or halted.</p> <p>0 An SDM mirror is accessed when the Shared Subsystem Area is addressed by the microengine.<br/>1 Enables microengine access to the Shared Subsystem Area.</p>                                                                                                                                                                                                                                                                                                                                                                           |
| 6<br>ETPD  | <p>Entry Table Pin Direction.</p> <p>This bit selects which channel signal, input or output, is used in the Entry Point selection. The ETPD value has to be compatible with the function chosen for the channel, selected in the field CFS. For details about Entry Table and condition encoding schemes, see <a href="#">Entry Points</a>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

*Table continues on the next page...*



**eTPU\_CCCR\_CnCR\_ENG1 field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | 0 Use PSTI for Entry Point selection.<br>1 Use PSTO for Entry Point selection.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 7<br>ETCS         | Entry Table Condition Select.<br><br>This bit determines the channel condition encoding scheme that selects, according to channel conditions, the Entry Point to be taken in an Entry Table. ETCS value has to be compatible with the function chosen for the channel, selected in field CFS. Two condition encoding schemes are available. For details about Entry Table and condition encoding schemes, see <a href="#">Entry Points</a> .<br><br>0 Select Standard Entry Table Condition encoding scheme.<br>1 Select Alternate Entry Table Condition encoding scheme. |
| 8–10<br>Reserved  | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 11–15<br>CFS      | Channel Function Select.<br><br>This field defines the function to be performed by the channel (see <a href="#">Functions and threads</a> ). The Function assigned to the channel has to be compatible with the channel condition encoding scheme, selected by field ETCS.                                                                                                                                                                                                                                                                                                |
| 16<br>ODIS        | Output Disable.<br><br>This bit enables the channel to have its output forced to the value opposite to OPOL when the output disable input signal corresponding to the channel group that it belongs is active (see <a href="#">eTPU channel output disable signals</a> and <a href="#">Figure 35-305</a> ).<br><br>0 Turns off the output disable feature for the channel.<br>1 Turns on the output disable feature for the channel.                                                                                                                                      |
| 17<br>OPOL        | Output Polarity.<br><br>Determines the output signal polarity. The activation of the output disable signal forces, when enabled by the ODIS bit, the channel output signal to the opposite of this polarity (see <a href="#">Figure 35-305</a> ).<br><br>0 Output active low (output disable drives output to high).<br>1 output active high (output disable drives output to low).                                                                                                                                                                                       |
| 18–20<br>Reserved | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 21–31<br>CPBA     | Channel n Parameter Base Address.<br><br>The value of this field times 8 specifies the SDM parameter base host (byte) address for channel n (2-parameter granularity; see <a href="#">SDM organization</a> ). As seen by the Host, the channel parameter base (byte) address is (without parameter sign extension) $eTPU\_Base + 0x8000 + CPBA * 8$ or (with parameter sign extension), $eTPU\_Base + 0xC000 + CPBA * 8$ .                                                                                                                                                |

35.13.2 eTPU Channel n Status Control Register for Engine 1 (eTPU\_CCCR\_CnSCR\_ENG1)

This register gathers the interrupt status bits of the channel and the Function Mode definition (read-write). Bits CIS, CIOS and DTRS for each channel can be also accessed from [eTPU Channel Interrupt Status Register for Engine 1 \(eTPU\\_GCR\\_CISR\\_ENG1\)](#), [eTPU Channel Interrupt Overflow Status Register for Engine 1 \(eTPU\\_GCR\\_CIOSR\\_ENG1\)](#) and [eTPU Channel Data Transfer Request Status Register for Engine 1 \(eTPU\\_GCR\\_CDTRSR\\_ENG1\)](#). Host must write 1 to clear a status bit.

Address: 0h base + 404h offset + (16d × i), where i=0d to 31d

|       |     |      |   |   |   |   |   |   |      |       |    |    |    |    |    |    |
|-------|-----|------|---|---|---|---|---|---|------|-------|----|----|----|----|----|----|
| Bit   | 0   | 1    | 2 | 3 | 4 | 5 | 6 | 7 | 8    | 9     | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | CIS | CLOS | 0 |   |   |   |   |   | DTRS | DTROS | 0  |    |    |    |    |    |
| W     | w1c | w1c  |   |   |   |   |   |   | w1c  | w1c   |    |    |    |    |    |    |
| Reset | 0   | 0    | 0 | 0 | 0 | 0 | 0 | 0 | 0    | 0     | 0  | 0  | 0  | 0  | 0  | 0  |

|       |     |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-----|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 16  | 17  | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | IPS | OPS | 0  | 0  |    |    |    |    |    |    |    |    |    |    |    | FM |
| W     |     |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0   | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

eTPU\_CCCR\_CnSCR\_ENG1 field descriptions

| Field    | Description                                                                                                                                                                                                                                                                                                                                                                                                                               |
|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>CIS | Channel Interrupt Status.<br><br>These bits are mirrored in <a href="#">eTPU Channel Interrupt Status Register for Engine 1 (eTPU_GCR_CISR_ENG1)</a> . Also, see <a href="#">Channel Interrupt and Data Transfer Requests</a> .<br><br>When read:<br>0.....Channel has no pending interrupt to the Host CPU.<br>1.....Channel has a pending interrupt to the Host CPU.<br><br>When written:<br>0.....Keep interrupt status bit unaltered. |

Table continues on the next page...

**eTPU\_CCCR\_CnSCR\_ENG1 field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | 1.....Clear interrupt status bit.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 1<br>CIOS         | <p>Channel Interrupt Overflow Status.</p> <p>These bits are mirrored in <a href="#">eTPU Channel Interrupt Overflow Status Register for Engine 1 (eTPU_GCR_CIOSR_ENG1)</a>. Also, see <a href="#">Interrupt and Data Transfer Request overflow</a>.</p> <p>When read:</p> <p>0.....Interrupt overflow negated for this channel.</p> <p>1.....Interrupt overflow asserted for this channel.</p> <p>When written:</p> <p>0.....Keep status bit unaltered.</p> <p>1.....Clear status bit.</p>                                           |
| 2–7<br>Reserved   | <p>Reserved.</p> <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 8<br>DTRS         | <p>Data Transfer Request Status.</p> <p>These bits are mirrored in <a href="#">eTPU Channel Data Transfer Request Status Register for Engine 1 (eTPU_GCR_CDTRSR_ENG1)</a>. Also, see <a href="#">Channel Interrupt and Data Transfer Requests</a>.</p> <p>When read:</p> <p>0.....Channel has no pending data transfer request.</p> <p>1.....Channel has a pending data transfer request.</p> <p>When written:</p> <p>0.....Keep status bit unaltered.</p> <p>1.....Clear status bit.</p>                                            |
| 9<br>DTROS        | <p>Data Transfer Request Overflow Status.</p> <p>These bits are mirrored in <a href="#">eTPU Channel Data Transfer Request Overflow Status Register for Engine 1 (eTPU_GCR_CDTROSR_ENG1)</a>. Also, see <a href="#">Interrupt and Data Transfer Request overflow</a>.</p> <p>When read:</p> <p>0.....Data transfer request overflow negated for this channel.</p> <p>1.....Data transfer request overflow asserted for this channel.</p> <p>When written:</p> <p>0.....Keep status bit unaltered.</p> <p>1.....Clear status bit.</p> |
| 10–15<br>Reserved | <p>Reserved.</p> <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 16<br>IPS         | <p>Channel Input Pin State.</p> <p>This bit shows the current value of the filtered channel input signal state.</p>                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 17<br>OPS         | Channel Output Pin State.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

*Table continues on the next page...*

**eTPU\_CCCR\_CnSCR\_ENG1 field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | This bit shows the current value driven in the channel output signal, including the effect of the external output disable feature (see <a href="#">eTPU channel output disable signals</a> ). If the channel input and output signals are connected to the same pad, OPS reflects the value driven to the pad. This is not necessarily the actual pad value, which drives the value in the bit IPS. |
| 18<br>Reserved    | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                            |
| 19–29<br>Reserved | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                            |
| 30–31<br>FM       | Channel Function Mode.<br><br>Each function uses this field for a specific configuration. These bits can be tested by microengine code (see <a href="#">Conditional/unconditional branch</a> ).<br><br>These bits are equivalent to the TPU/TPU2/TPU3 Host Sequence (HSQ) bits.                                                                                                                     |

**35.13.3 eTPU Channel n Host Service Request Register for Engine 1 (eTPU\_CCCR\_CnHSRR\_ENG1)**

This register is used by the Host to issue service requests to the channel.

Address: 0h base + 408h offset + (16d × i), where i=0d to 31d

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28  | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | HSR |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0  | 0  |    |

**eTPU\_CCCR\_CnHSRR\_ENG1 field descriptions**

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–28<br>Reserved | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 29–31<br>HSR     | Host Service Request<br><br>This field is used by the Host CPU to request service to the channel (see <a href="#">Host Service Requests</a> ).<br><br>HSR value turns to 000 automatically at the end of microengine service for that channel, but only if the thread started due to an HSR. Host should write HSR greater than 0 only when HSR=0. Writing HSR=000 withdraws a pending request if scheduler did not begin to resolve the Entry Point yet, but it does not abort the service thread from that point on. For more details, see <a href="#">Entry Points</a> and <a href="#">Host Service Requests</a> .<br><br>000 No Host Service Request pending. |

*Table continues on the next page...*

**eTPU\_CCCR\_CnHSRR\_ENG1 field descriptions (continued)**

| Field | Description                                      |
|-------|--------------------------------------------------|
| 001   | Function-dependent Host Service Request pending. |
| ---   | ---                                              |
| 111   | Function-dependent Host Service Request pending. |

**35.13.4 eTPU Channel n Configuration Register for Engine 2 (eTPU\_CCCR\_CnCR\_ENG2)**

This register gathers configurations set individually per channel.

Address: 0h base + 800h offset + (16d × i), where i=0d to 31d

|       |     |      |     |   |       |       |      |      |   |   |    |    |    |    |    |    |
|-------|-----|------|-----|---|-------|-------|------|------|---|---|----|----|----|----|----|----|
| Bit   | 0   | 1    | 2   | 3 | 4     | 5     | 6    | 7    | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     |     |      |     |   |       |       |      |      |   | 0 |    |    |    |    |    |    |
| W     | CIE | DTRE | CPR |   | T4DIS | SSMAE | ETPD | ETCS |   |   |    |    |    |    |    |    |
| Reset | 0   | 0    | 0   | 0 | 0     | 0     | 0    | 0    | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |      |      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|------|------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 16   | 17   | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |      |      |    |    | 0  |    |    |    |    |    |    |    |    |    |    |    |
| W     | ODIS | OPOL |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0    | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**eTPU\_CCCR\_CnCR\_ENG2 field descriptions**

| Field     | Description                                                                                                                                                                                                                                                                                                                                                     |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>CIE  | Channel Interrupt Enable (see <a href="#">Channel Interrupt and Data Transfer Requests</a> ).<br><br>This bit is mirrored from <a href="#">eTPU Channel Interrupt Enable Register for Engine 2 (eTPU_GCR_CIER_ENG2)</a> .<br><br>0 Disable interrupt for this channel.<br>1 Enable interrupt for this channel.                                                  |
| 1<br>DTRE | Channel Data Transfer Request Enable (see <a href="#">Channel Interrupt and Data Transfer Requests</a> ).<br><br>This bit is mirrored from <a href="#">eTPU Channel Data Transfer Request Enable Register for Engine 2 (eTPU_GCR_CDTRE_ENG2)</a> .<br><br>0 Disable data transfer request for this channel.<br>1 Enable data transfer request for this channel. |

Table continues on the next page...

**eTPU\_CCCR\_CnCR\_ENG2 field descriptions (continued)**

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2–3<br>CPR       | <p>Channel Priority.</p> <p>This field defines the priority level for the channel, used by the Hardware Scheduler (see <a href="#">Scheduler</a>).</p> <p>00 Disabled<br/>01 Low<br/>10 Middle<br/>11 High</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 4<br>T4DIS       | <p>T4 channel action Disable.</p> <p>This bit forces the channel to work on T2 timing mode, regardless of the FCSS and CDFC fields in <a href="#">eTPU Engine 2 Configuration Register (eTPU_SCR_ECR_ENG2)</a>, or of the TCR1CS field in <a href="#">eTPU Engine 2 Time Base Configuration Register (eTPU_TBR_TBCR_ENG2)</a>. For more details, see <a href="#">Channel timing modes</a>.</p> <p>0 Channel works either in T2 or T2/T4 timing modes, depending on the TCR1CS field in <a href="#">eTPU Engine 2 Time Base Configuration Register (eTPU_TBR_TBCR_ENG2)</a> and the FCSS and CDFC fields in <a href="#">eTPU Engine 2 Configuration Register (eTPU_SCR_ECR_ENG2)</a>.</p> <p>1 Channel works in T2 timing mode, regardless of the TCR1CS field in <a href="#">eTPU Engine 2 Time Base Configuration Register (eTPU_TBR_TBCR_ENG2)</a> or the FCSS and CDFC fields in <a href="#">eTPU Engine 2 Configuration Register (eTPU_SCR_ECR_ENG2)</a>.</p> |
| 5<br>SSMAE       | <p>Shared Subsystem Microengine Access Enable.</p> <p>This bit enables the microengine access to the Shared Subsystem Area (see <a href="#">Shared Subsystems support</a>). When this bit is negated, DIOB address values pointing to the Shared Subsystem Area actually access SDM mirrors, in legacy mode.</p> <p><b>NOTE:</b> SSMAE value must only be changed when the engine is either stopped or halted.</p> <p>0 An SDM mirror is accessed when the Shared Subsystem Area is addressed by the microengine.<br/>1 Enables microengine access to the Shared Subsystem Area.</p>                                                                                                                                                                                                                                                                                                                                                                              |
| 6<br>ETPD        | <p>Entry Table Pin Direction.</p> <p>This bit selects which channel signal, input or output, is used in the Entry Point selection. The ETPD value has to be compatible with the function chosen for the channel, selected in the field CFS. For details about Entry Table and condition encoding schemes, see <a href="#">Entry Points</a>.</p> <p>0 Use PSTI for Entry Point selection.<br/>1 Use PSTO for Entry Point selection.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 7<br>ETCS        | <p>Entry Table Condition Select.</p> <p>This bit determines the channel condition encoding scheme that selects, according to channel conditions, the Entry Point to be taken in an Entry Table. ETCS value has to be compatible with the function chosen for the channel, selected in field CFS. Two condition encoding schemes are available. For details about Entry Table and condition encoding schemes, see <a href="#">Entry Points</a>.</p> <p>0 Select Standard Entry Table Condition encoding scheme.<br/>1 Select Alternate Entry Table Condition encoding scheme.</p>                                                                                                                                                                                                                                                                                                                                                                                  |
| 8–10<br>Reserved | <p>Reserved.</p> <p>This field is reserved.<br/>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

*Table continues on the next page...*

**eTPU\_CCCR\_CnCR\_ENG2 field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 11–15<br>CFS      | Channel Function Select.<br><br>This field defines the function to be performed by the channel (see <a href="#">Functions and threads</a> ). The Function assigned to the channel has to be compatible with the channel condition encoding scheme, selected by field ETCS.                                                                                                                                                           |
| 16<br>ODIS        | Output Disable.<br><br>This bit enables the channel to have its output forced to the value opposite to OPOL when the output disable input signal corresponding to the channel group that it belongs is active (see <a href="#">eTPU channel output disable signals</a> and <a href="#">Figure 35-305</a> ).<br><br>0 Turns off the output disable feature for the channel.<br>1 Turns on the output disable feature for the channel. |
| 17<br>OPOL        | Output Polarity.<br><br>Determines the output signal polarity. The activation of the output disable signal forces, when enabled by the ODIS bit, the channel output signal to the opposite of this polarity (see <a href="#">Figure 35-305</a> ).<br><br>0 Output active low (output disable drives output to high).<br>1 output active high (output disable drives output to low).                                                  |
| 18–20<br>Reserved | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                             |
| 21–31<br>CPBA     | Channel n Parameter Base Address.<br><br>The value of this field times 8 specifies the SDM parameter base host (byte) address for channel n (2-parameter granularity; see <a href="#">SDM organization</a> ). As seen by the Host, the channel parameter base (byte) address is (without parameter sign extension) $eTPU\_Base + 0x8000 + CPBA * 8$ or (with parameter sign extension), $eTPU\_Base + 0xC000 + CPBA * 8$ .           |

35.13.5 eTPU Channel n Status Control Register for Engine 2 (eTPU\_CCCR\_CnSCR\_ENG2)

This register gathers the interrupt status bits of the channel and the Function Mode definition (read-write). Bits CIS, CIOS and DTRS for each channel can be also accessed from [eTPU Channel Interrupt Status Register for Engine 2 \(eTPU\\_GCR\\_CISR\\_ENG2\)](#), [eTPU Channel Interrupt Overflow Status Register for Engine 2 \(eTPU\\_GCR\\_CIOSR\\_ENG2\)](#) and [eTPU Channel Data Transfer Request Status Register for Engine 2 \(eTPU\\_GCR\\_CDTRSR\\_ENG2\)](#). Host must write 1 to clear a status bit.

Address: 0h base + 804h offset + (16d × i), where i=0d to 31d

|       |     |      |   |   |   |   |   |   |      |       |    |    |    |    |    |    |
|-------|-----|------|---|---|---|---|---|---|------|-------|----|----|----|----|----|----|
| Bit   | 0   | 1    | 2 | 3 | 4 | 5 | 6 | 7 | 8    | 9     | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | CIS | CLOS | 0 |   |   |   |   |   | DTRS | DTROS | 0  |    |    |    |    |    |
| W     | w1c | w1c  |   |   |   |   |   |   | w1c  | w1c   |    |    |    |    |    |    |
| Reset | 0   | 0    | 0 | 0 | 0 | 0 | 0 | 0 | 0    | 0     | 0  | 0  | 0  | 0  | 0  | 0  |

|       |     |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-----|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 16  | 17  | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | IPS | OPS | 0  | 0  |    |    |    |    |    |    |    |    |    |    |    | FM |
| W     |     |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0   | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

eTPU\_CCCR\_CnSCR\_ENG2 field descriptions

| Field    | Description                                                                                                                                                                                                                                                                                                                                                                                                                               |
|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>CIS | Channel Interrupt Status.<br><br>These bits are mirrored in <a href="#">eTPU Channel Interrupt Status Register for Engine 2 (eTPU_GCR_CISR_ENG2)</a> . Also, see <a href="#">Channel Interrupt and Data Transfer Requests</a> .<br><br>When read:<br>0.....Channel has no pending interrupt to the Host CPU.<br>1.....Channel has a pending interrupt to the Host CPU.<br><br>When written:<br>0.....Keep interrupt status bit unaltered. |

Table continues on the next page...



**eTPU\_CCCR\_CnSCR\_ENG2 field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | 1.....Clear interrupt status bit.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 1<br>CIOS         | <p>Channel Interrupt Overflow Status.</p> <p>These bits are mirrored in <a href="#">eTPU Channel Interrupt Overflow Status Register for Engine 2 (eTPU_GCR_CIOSR_ENG2)</a>. Also, see <a href="#">Interrupt and Data Transfer Request overflow</a>.</p> <p>When read:</p> <p>0.....Interrupt overflow negated for this channel.</p> <p>1.....Interrupt overflow asserted for this channel.</p> <p>When written:</p> <p>0.....Keep status bit unaltered.</p> <p>1.....Clear status bit.</p>                                           |
| 2–7<br>Reserved   | <p>Reserved.</p> <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 8<br>DTRS         | <p>Data Transfer Request Status.</p> <p>These bits are mirrored in <a href="#">eTPU Channel Interrupt Status Register for Engine 2 (eTPU_GCR_CISR_ENG2)</a>. Also, see <a href="#">Channel Interrupt and Data Transfer Requests</a>.</p> <p>When read:</p> <p>0.....Channel has no pending data transfer request.</p> <p>1.....Channel has a pending data transfer request.</p> <p>When written:</p> <p>0.....Keep status bit unaltered.</p> <p>1.....Clear status bit.</p>                                                          |
| 9<br>DTROS        | <p>Data Transfer Request Overflow Status.</p> <p>These bits are mirrored in <a href="#">eTPU Channel Data Transfer Request Overflow Status Register for Engine 2 (eTPU_GCR_CDTROSR_ENG2)</a>. Also, see <a href="#">Channel Interrupt and Data Transfer Requests</a>.</p> <p>When read:</p> <p>0.....Data transfer request overflow negated for this channel.</p> <p>1.....Data transfer request overflow asserted for this channel.</p> <p>When written:</p> <p>0.....Keep status bit unaltered.</p> <p>1.....Clear status bit.</p> |
| 10–15<br>Reserved | <p>Reserved.</p> <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 16<br>IPS         | <p>Channel Input Pin State.</p> <p>This bit shows the current value of the filtered channel input signal state.</p>                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 17<br>OPS         | Channel Output Pin State.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

*Table continues on the next page...*

**eTPU\_CCCR\_CnSCR\_ENG2 field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | This bit shows the current value driven in the channel output signal, including the effect of the external output disable feature (see <a href="#">eTPU channel output disable signals</a> ). If the channel input and output signals are connected to the same pad, OPS reflects the value driven to the pad. This is not necessarily the actual pad value, which drives the value in the bit IPS. |
| 18<br>Reserved    | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                            |
| 19–29<br>Reserved | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                            |
| 30–31<br>FM       | Channel Function Mode.<br><br>Each function uses this field for a specific configuration. These bits can be tested by microengine code (see <a href="#">Conditional/unconditional branch</a> ).<br><br>These bits are equivalent to the TPU/TPU2/TPU3 Host Sequence (HSQ) bits.                                                                                                                     |

**35.13.6 eTPU Channel n Host Service Request Register for Engine 2 (eTPU\_CCCR\_CnHSRR\_ENG2)**

This register is used by the Host to issue service requests to the channel.

Address: 0h base + 808h offset + (16d × i), where i=0d to 31d

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28  | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | HSR |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0  | 0  |    |

**eTPU\_CCCR\_CnHSRR\_ENG2 field descriptions**

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–28<br>Reserved | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 29–31<br>HSR     | Host Service Request<br><br>This field is used by the Host CPU to request service to the channel (see <a href="#">Host Service Requests</a> ).<br><br>HSR value turns to 000 automatically at the end of microengine service for that channel, but only if the thread started due to an HSR. Host should write HSR greater than 0 only when HSR=0. Writing HSR=000 withdraws a pending request if scheduler did not begin to resolve the Entry Point yet, but it does not abort the service thread from that point on. For more details, see <a href="#">Entry Points</a> and <a href="#">Host Service Requests</a> .<br><br>000 No Host Service Request pending. |

*Table continues on the next page...*

**eTPU\_CCCR\_CnHSRR\_ENG2 field descriptions (continued)**

| Field | Description                                      |
|-------|--------------------------------------------------|
| 001   | Function-dependent Host Service Request pending. |
| ---   | ---                                              |
| 111   | Function-dependent Host Service Request pending. |

## 35.14 Shared Subsystem Integration memory map and registers

Shared Subsystems Integration Registers configure aspects of how the shared modules interface with the eTPU. There is one integration register for each subsystem unit, which can be, depending on the specific subsystem, a whole IP module, a sub-module, or a channel within a module. For more details, see [Shared Subsystems support](#).

**eTPU\_SSIR memory map**

| Address offset (hex) | Register name                                                       | Width (in bits) | Access | Reset value | Section/page                 |
|----------------------|---------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| C00                  | eTPU Shared Subsystem n Integration Register (eTPU_SSIR_ETPUSSSIR0) | 32              | R/W    | 0000_0000h  | <a href="#">35.14.1/1400</a> |
| C04                  | eTPU Shared Subsystem n Integration Register (eTPU_SSIR_ETPUSSSIR1) | 32              | R/W    | 0000_0000h  | <a href="#">35.14.1/1400</a> |
| C08                  | eTPU Shared Subsystem n Integration Register (eTPU_SSIR_ETPUSSSIR2) | 32              | R/W    | 0000_0000h  | <a href="#">35.14.1/1400</a> |
| C0C                  | eTPU Shared Subsystem n Integration Register (eTPU_SSIR_ETPUSSSIR3) | 32              | R/W    | 0000_0000h  | <a href="#">35.14.1/1400</a> |
| C10                  | eTPU Shared Subsystem n Integration Register (eTPU_SSIR_ETPUSSSIR4) | 32              | R/W    | 0000_0000h  | <a href="#">35.14.1/1400</a> |
| C14                  | eTPU Shared Subsystem n Integration Register (eTPU_SSIR_ETPUSSSIR5) | 32              | R/W    | 0000_0000h  | <a href="#">35.14.1/1400</a> |
| C18                  | eTPU Shared Subsystem n Integration Register (eTPU_SSIR_ETPUSSSIR6) | 32              | R/W    | 0000_0000h  | <a href="#">35.14.1/1400</a> |
| C1C                  | eTPU Shared Subsystem n Integration Register (eTPU_SSIR_ETPUSSSIR7) | 32              | R/W    | 0000_0000h  | <a href="#">35.14.1/1400</a> |
| C20                  | eTPU Shared Subsystem n Integration Register (eTPU_SSIR_ETPUSSSIR8) | 32              | R/W    | 0000_0000h  | <a href="#">35.14.1/1400</a> |
| C24                  | eTPU Shared Subsystem n Integration Register (eTPU_SSIR_ETPUSSSIR9) | 32              | R/W    | 0000_0000h  | <a href="#">35.14.1/1400</a> |

### 35.14.1 eTPU Shared Subsystem n Integration Register (eTPU\_SSIR\_ETPUSSSIRn)

This register is used to configure the interface of a subsystem unit with the eTPU.

Address: 0h base + C00h offset + (4d × i), where i=0d to 9d

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |    |    |    |    |    |    |    |      |    |    |       |    |    |    |    |
|-------|----|----|----|----|----|----|----|----|------|----|----|-------|----|----|----|----|
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24   | 25 | 26 | 27    | 28 | 29 | 30 | 31 |
| R     | 0  |    |    |    |    |    |    |    | LENG |    | 0  | LCHAN |    |    |    |    |
| W     |    |    |    |    |    |    |    |    |      |    |    |       |    |    |    |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0  | 0  | 0     | 0  | 0  | 0  | 0  |

eTPU\_SSIR\_ETPUSSSIRn field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–23<br>Reserved | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                     |
| 24–25<br>LENG    | Link Engine.<br><br>LENG specifies the engine that receives a Link Service Request when an interrupt is issued by the Shared Unit, or disables Shared Unit linking. The channel that receives the Link Service Request is specified in the LCHAN field.<br><br>00 Shared Unit Link Service Request disabled<br>01 Engine 1 receives Shared Unit Link Service Request<br>10 Engine 2 receives Shared Unit Link Service Request<br>11 Reserved |
| 26<br>Reserved   | Reserved.<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                     |
| 27–31<br>LCHAN   | Link Channel.<br><br>LCHAN specifies the channel linked when an interrupt is issued by the Shared Unit, if enabled in the LENG field. The field is meaningless if LENG disables Shared Unit linking. For more information on Link Service Requests, see <a href="#">Channel Link</a> .                                                                                                                                                       |

## 35.15 Functional description

### 35.15.1 Functions and threads

eTPU processing is event-driven, in the sense that eTPU microcode only runs to service a request from an event. Service Requests may result from the occurrence of any of the following events:

- Host CPU writing a non-zero value to the channel HSR (Host Service Request) field in [eTPU Channel n Host Service Request Register for Engine 1 \(eTPU\\_CCCR\\_CnHSRR\\_ENG1\)](#) and [eTPU Channel n Host Service Request Register for Engine 2 \(eTPU\\_CCCR\\_CnHSRR\\_ENG2\)](#).
- Occurrence of a time base match, an input signal transition, or a specific combination of them (depending on the Channel Mode currently configured).
- A Link Service Request.

A given event is always associated to only one Channel:

- There is one HSR register field for each Channel
- Each signal is associated with only one Channel, which has its own Match registers and independent mode configuration.
- Each Link Service Request can have only one Channel as a target.

Service Request processing is done by a set of microengine routines. A set of related routines that implement a specific channel application is called a **Function**. One or more Functions reside on SCM, limited only by the SCM space available, size of microcode Functions and the number of entry points available. Each Engine can be controlled by up to 32 Functions at a time.

A Function can be assigned to several channels, but only one Function can be assigned to a given Channel at a time. This is defined by the Host through the Channel Configuration Registers (see [Channel Configuration Control memory map and registers](#)).

The term **Thread** will be used hereafter to refer to a service routine of a Function, or its execution. A Thread is constructed of a specific number of microinstructions, typically the code necessary to calculate the next phase of waveform to be input to, or output from, a given channel. Once a Thread begins, its execution cannot be interrupted. A Thread normally finishes when an END microinstruction is executed.

A given Thread is selected and called by the Scheduler depending on the following:

- the type of event that generated the service request.

## Functional description

- the Function assigned to the target channel.
- target channel pin state.
- the state of the channel logic.
- the priority assigned to the target channel, relative to the priorities of other channels with pending service requests

The mechanism to select a thread based on the channel Function and type of event is described in the [Entry Points](#).

The priority mechanism that determines the order of Thread execution amongst pending service requests is described in [Scheduler](#).

### 35.15.1.1 Entry Points

#### 35.15.1.1.1 Entry Table

Each Thread has its **Entry Point**, which contains the SCM address of its first instruction, besides other information. For a complete Entry Point description, see [Entry Point format](#).

Once the Scheduler chooses a channel among pending Service Requests, the Entry Point is taken from an **Entry Table**, based on the Function assigned for the channel and other conditions. Entry Table layout is shown in the following figure.

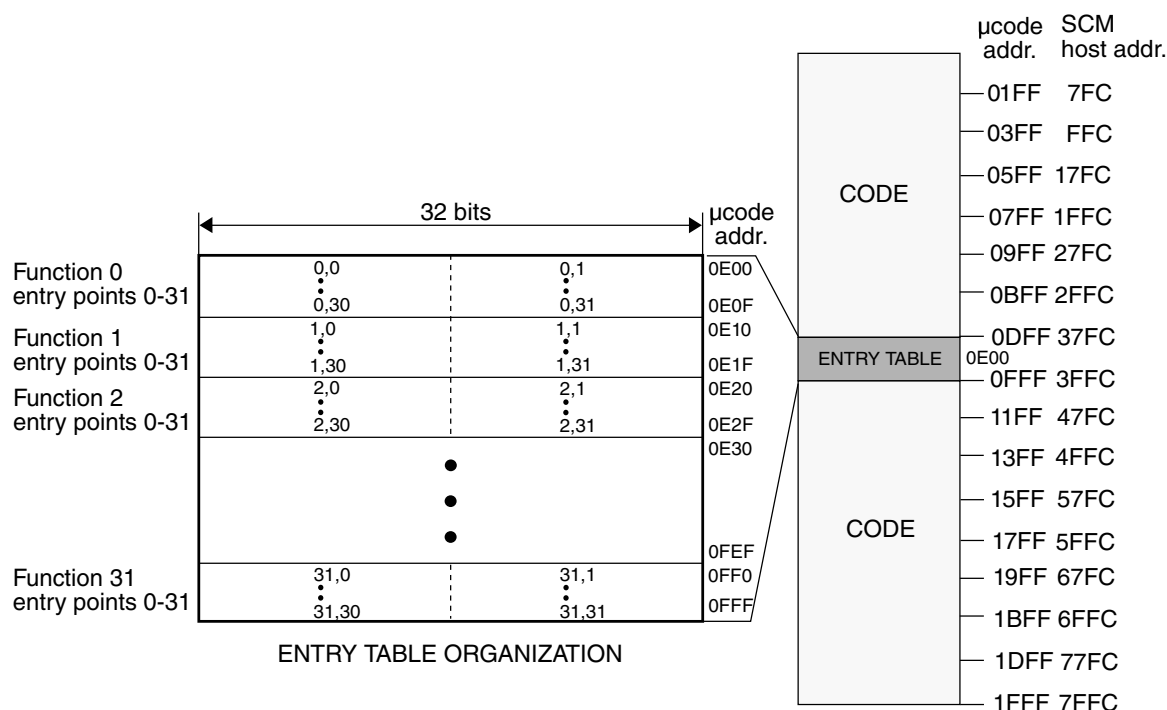


Figure 35-291. Entry Table

The Entry Table is organized by Functions. Each Function can have up to 32 Entry Points of 16 bits each, corresponding to 32 possible Threads per Function. Each Entry Point location in the table corresponds to a combination of events and channel states (see [Entry Point address generation](#)). A single Thread can be associated to more than one combination, having its Entry Point repeated in the table. Each 32-bit word in the Entry Table holds two Entry Points.

Note that the Entry Table can be placed in any SCM address multiple of the Entry Table size, determined by the ETB field in [eTPU Engine 1 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG1\)](#) and [eTPU Engine 2 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG2\)](#). However, it is recommended to place the Entry Table at the start of the SCM to get continuous code memory and to ease the eventual migration of the code from larger parts down to smaller ones without rearranging the binary image, but this is not a restriction. Unused Entry Points may be used for microcode, so this organization extends the microcode continuous area to the unused area of the Entry Table. For this purpose, Function numbers should be selected from 0 up to 31. If, for example, only 8 functions are implemented, only the Entry Table locations for Functions 0 to 7 are used, and the Entry Table locations for functions 8 to 31 can be used as microinstruction memory (adding extra continuous 1536 bytes for microprogram usage).

One way of implementing different sets of Functions is having more than one Entry Table, and configuring the eTPU with the appropriate one for the application by changing the ETB field in [eTPU Engine 1 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG1\)](#) and [eTPU Engine 2 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG2\)](#). Note that the Engines can use different Entry Tables, with or without the same set of Functions.

### 35.15.1.1.2 Entry Point address generation

The Entry Point address within the Entry Table is determined by the Function assigned to the Channel, the state of the Channel, the type of event, and the condition encoding scheme. Together with the Entry Table base address, they form the Entry Point Address at the SCM, as shown in the following figure.

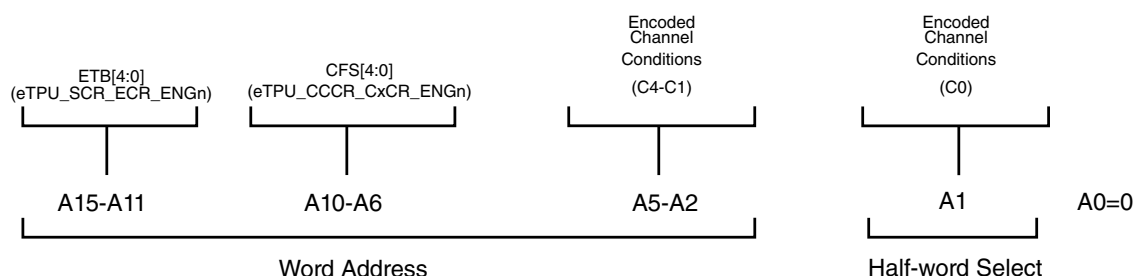


**Figure 35-292. Entry Point Address (host address offset)**

The type of event and channel state are coded in the Encoded Channel Conditions field C4-C0, according to one of two encoding schemes:

- Standard Entry Table Condition encoding scheme, shown in [Table 35-333](#), which privileges Host Service Requests.
- Alternate Entry Table Condition encoding scheme, shown in [Table 35-334](#), which focus on other events and state decoding.

The events that take part on condition encoding generate a Service Request, and have four origins:

1. Match Recognition (caused by greater/equal match, or equal-only, between the value TCR1/2 and the value stored in the channel match registers). eTPU channels support single and double match in various modes of match recognition; see [Match Recognition](#).
2. Transition Detect Service Request (channel input signal transition detection of a selected edge). The eTPU channels support single and double transition, which together with the double match options provide various modes of transition detection; see [Transition detection and Time Base Capture](#).



3. Channel Linking Service Request (microcode writing the channel number to the LINK register). Link service request allows one channel to activate another (see [Channel Link](#)). Links may also come from Shared Subsystem units (see [Shared Subsystems support](#)).
4. Host Service Request (Host writes a non-zero value to the HSR bits of the channel; see [Host Service Requests](#)).

### Note

Even if a Transition or Match Service Request is inhibited (by channel mode/state or SRI), the Transition Detection and Match Recognition are taken into account for condition encoding. That is, the MRLA/B and TDLA/B flags are used, not their respective Service Requests.

Columns Host Request Bits, Link Request, MatchA/TransB, and MatchB/TransA determine the type of event. A non-zero value in these columns represents the recognition of the event, while "x" indicates that its recognition is irrelevant. Values 1 and 0 mean that event was recognized or not, respectively. Note that Match and Transition events may occur and not be recognized, and in this case it assumes value 0 for the condition encoding. The recognition of such an occurred event depends on the channel mode assigned and other conditions, as described in [Enhanced Channels](#).

The **Host Service Request Bits** column refers to the value written by the Host CPU to the Host Service Request Registers ([eTPU Channel n Host Service Request Register for Engine 1 \(eTPU\\_CCCR\\_CnHSRR\\_ENG1\)](#) and [eTPU Channel n Host Service Request Register for Engine 2 \(eTPU\\_CCCR\\_CnHSRR\\_ENG2\)](#)) of the Channel being serviced. Note that the bits on this row are coded (3-bit representation). If the value of HSR is not zero, then the Host actually requested service.

The **Link Request** column refers to the occurrence of a Channel Link or SSS Link requests.

The **MatchA/TransB** column refers to the recognition of either a Match event specified by MatchA channel register or the detection of a channel input signal event specified by the IPACB configuration register (see [Pin Control Registers](#)).

The **MatchB/TransA** column refers to the recognition of either a Match event specified by MatchB channel register or the detection of a channel input signal event specified by the IPACA configuration register (see [Pin Control Registers](#)).

For the channel input signal, MatchA and MatchB provide double timeout conditions which depend on the channel mode programming (see [Channel Modes](#)). If the channel is used for output only, there are no transition detections, so the MatchB/TransA column

represents only Match B, and MatchA/TransB column the Match A. In this case Match A and Match B are separated to give better state resolution in double match output functions. For more information about channel requests refer to [Enhanced Channels](#).

Besides those events, the following channel state conditions help to determine the Entry Point:

1. Channel Flags 0 and 1: these are channel-internal flags (not in SDM) associated with a channel. Their values are set by microcode (see [Channel flags operations](#)).
2. Input Pin state or Output Flip Flop: the state (0 or 1) of the channel input signal after the Enhanced Filter (see [Enhanced Digital Filter - EDF](#)), or the state driven to the output signal. Which one (input or output) is used is selected by the ETPD field in [eTPU Channel n Configuration Register for Engine 1 \(eTPU\\_CCCR\\_CnCR\\_ENG1\)](#) and [eTPU Channel n Configuration Register for Engine 2 \(eTPU\\_CCCR\\_CnCR\\_ENG2\)](#).

The two Entry Table Condition encoding schemes combine events and state conditions differently, as detailed in the following sections.

### 35.15.1.1.3 Standard condition encoding scheme

In this scheme, shown in the following table, all 7 HSR combinations are used and other event type columns are marked "x" when HSR is non-zero, indicating that Host Service Request has priority over any other type of event. However, when an HSR service thread is called (entry numbers 0 to 9), other events may also have been recognized, and it is microcode responsibility to check them.

When HSR is 0, i.e., Host did not issue a Service Request to the channel, the other event conditions, the input signal state and channel flags determine the Entry Point. Note that channel flag 1 does not influence the encoding in this scheme.

**Table 35-333. Standard channel condition encoding scheme**

| No. | Encoded channel conditions [C4-C0] | Host Service Request bits | Link request | MatchA/ TransB | MatchB/ TransA | In/output pin state <sup>1</sup> | Channel Flag1 | Channel Flag0 |
|-----|------------------------------------|---------------------------|--------------|----------------|----------------|----------------------------------|---------------|---------------|
| 0   | 001                                | 00000                     | x            | x              | x              | 0                                | x             | 0             |
| 1   | 00001                              | 001                       | x            | x              | x              | 0                                | x             | 1             |
| 2   | 00010                              | 001                       | x            | x              | x              | 1                                | x             | 0             |
| 3   | 00011                              | 001                       | x            | x              | x              | 1                                | x             | 1             |
| 4   | 00100                              | 010                       | x            | x              | x              | x                                | x             | x             |
| 5   | 00101                              | 011                       | x            | x              | x              | x                                | x             | x             |
| 6   | 00110                              | 100                       | x            | x              | x              | x                                | x             | x             |

*Table continues on the next page...*

**Table 35-333. Standard channel condition encoding scheme (continued)**

| No.                                                                                                                                                                                     | Encoded channel conditions [C4-C0] | Host Service Request bits | Link request | MatchA/TransB | MatchB/TransA | In/output pin state <sup>1</sup> | Channel Flag1 | Channel Flag0 |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------|---------------------------|--------------|---------------|---------------|----------------------------------|---------------|---------------|
| 7                                                                                                                                                                                       | 00111                              | 101                       | x            | x             | x             | x                                | x             | x             |
| 8                                                                                                                                                                                       | 01000                              | 110                       | x            | x             | x             | x                                | x             | x             |
| 9                                                                                                                                                                                       | 01001                              | 111                       | x            | x             | x             | x                                | x             | x             |
| 10                                                                                                                                                                                      | 01010                              | 000                       | 1            | 1             | 1             | x                                | x             | 0             |
| 11                                                                                                                                                                                      | 01011                              | 000                       | 1            | 1             | 1             | x                                | x             | 1             |
| 12                                                                                                                                                                                      | 01100                              | 000                       | 0            | 0             | 1             | 0                                | x             | 0             |
| 13                                                                                                                                                                                      | 01101                              | 000                       | 0            | 0             | 1             | 0                                | x             | 1             |
| 14                                                                                                                                                                                      | 01110                              | 000                       | 0            | 0             | 1             | 1                                | x             | 0             |
| 15                                                                                                                                                                                      | 01111                              | 000                       | 0            | 0             | 1             | 1                                | x             | 1             |
| 16                                                                                                                                                                                      | 10000                              | 000                       | 0            | 1             | 0             | 0                                | x             | 0             |
| 17                                                                                                                                                                                      | 10001                              | 000                       | 0            | 1             | 0             | 0                                | x             | 1             |
| 18                                                                                                                                                                                      | 10010                              | 000                       | 0            | 1             | 0             | 1                                | x             | 0             |
| 19                                                                                                                                                                                      | 10011                              | 000                       | 0            | 1             | 0             | 1                                | x             | 1             |
| 20                                                                                                                                                                                      | 10100                              | 000                       | 0            | 1             | 1             | 0                                | x             | 0             |
| 21                                                                                                                                                                                      | 10101                              | 000                       | 0            | 1             | 1             | 0                                | x             | 1             |
| 22                                                                                                                                                                                      | 10110                              | 000                       | 0            | 1             | 1             | 1                                | x             | 0             |
| 23                                                                                                                                                                                      | 10111                              | 000                       | 0            | 1             | 1             | 1                                | x             | 1             |
| 24                                                                                                                                                                                      | 11000                              | 000                       | 1            | 0             | 0             | 0                                | x             | 0             |
| 25                                                                                                                                                                                      | 11001                              | 000                       | 1            | 0             | 0             | 0                                | x             | 1             |
| 26                                                                                                                                                                                      | 11010                              | 000                       | 1            | 0             | 0             | 1                                | x             | 0             |
| 27                                                                                                                                                                                      | 11011                              | 000                       | 1            | 0             | 0             | 1                                | x             | 1             |
| 28                                                                                                                                                                                      | 11100                              | 000                       | 1            | 0             | 1             | x                                | x             | 0             |
| 29                                                                                                                                                                                      | 11101                              | 000                       | 1            | 0             | 1             | x                                | x             | 1             |
| 30                                                                                                                                                                                      | 11110                              | 000                       | 1            | 1             | 0             | x                                | x             | 0             |
| 31                                                                                                                                                                                      | 11111                              | 000                       | 1            | 1             | 0             | x                                | x             | 1             |
| <div style="display: flex; align-items: center;"> <div style="width: 100px; height: 30px; background-color: #cccccc; margin-right: 10px;"></div> <div>Host Service Request</div> </div> |                                    |                           |              |               |               |                                  |               |               |

1. The ETPD field in [CCR\\_ENG1](#) and [CCR\\_ENG2](#) selects between input and output pin state.

#### 35.15.1.1.4 Alternate condition encoding scheme

This scheme is shown in the following table. Because the HSR bits cannot be tested by microcode, only three distinct Host Service Request can be used:

1. HSR=010 or 011, which are coded into the same Entry Points (0 to 3)
2. HSR=100, 101 or 001, which are all coded into Entry Point 4

### 3. HSR=110 or 111, which are both coded into Entry Point 5

The remaining Entry Points use both channel flags for better state decoding, making this scheme better suited for Functions which need more states and/or faster state decoding, without needing many HSRs.

**Table 35-334. Alternate channel condition encoding scheme**

| No. | Encoded channel conditions [C4-C0] | Host Service Request bits | Link request | MatchA/TransB | MatchB/TransA | In/Output pin state <sup>1</sup> | Channel Flag1 | Channel Flag0 |
|-----|------------------------------------|---------------------------|--------------|---------------|---------------|----------------------------------|---------------|---------------|
| 0   | 00000                              | 01x                       | x            | x             | x             | 0                                | x             | 0             |
| 1   | 00001                              | 01x                       | x            | x             | x             | 0                                | x             | 1             |
| 2   | 00010                              | 01x                       | x            | x             | x             | 1                                | x             | 0             |
| 3   | 00011                              | 01x                       | x            | x             | x             | 1                                | x             | 1             |
| 4   | 00100                              | 10x/001                   | x            | x             | x             | x                                | x             | x             |
| 5   | 00101                              | 11x                       | x            | x             | x             | x                                | x             | x             |
| 6   | 00110                              | 000                       | 1            | 0             | 0             | 0                                | x             | x             |
| 7   | 00111                              | 000                       | 1            | 0             | 0             | 1                                | x             | x             |
| 8   | 01000                              | 000                       | x            | 1             | 0             | 0                                | 0             | 0             |
| 9   | 01001                              | 000                       | x            | 1             | 0             | 0                                | 0             | 1             |
| 10  | 01010                              | 000                       | x            | 1             | 0             | 0                                | 1             | 0             |
| 11  | 01011                              | 000                       | x            | 1             | 0             | 0                                | 1             | 1             |
| 12  | 01100                              | 000                       | x            | 1             | 0             | 1                                | 0             | 0             |
| 13  | 01101                              | 000                       | x            | 1             | 0             | 1                                | 0             | 1             |
| 14  | 01110                              | 000                       | x            | 1             | 0             | 1                                | 1             | 0             |
| 15  | 01111                              | 000                       | x            | 1             | 0             | 1                                | 1             | 1             |
| 16  | 10000                              | 000                       | x            | 0             | 1             | 0                                | 0             | 0             |
| 17  | 10001                              | 000                       | x            | 0             | 1             | 0                                | 0             | 1             |
| 18  | 10010                              | 000                       | x            | 0             | 1             | 0                                | 1             | 0             |
| 19  | 10011                              | 000                       | x            | 0             | 1             | 0                                | 1             | 1             |
| 20  | 10100                              | 000                       | x            | 0             | 1             | 1                                | 0             | 0             |
| 21  | 10101                              | 000                       | x            | 0             | 1             | 1                                | 0             | 1             |
| 22  | 10110                              | 000                       | x            | 0             | 1             | 1                                | 1             | 0             |
| 23  | 10111                              | 000                       | x            | 0             | 1             | 1                                | 1             | 1             |
| 24  | 11000                              | 000                       | x            | 1             | 1             | 0                                | 0             | 0             |
| 25  | 11001                              | 000                       | x            | 1             | 1             | 0                                | 0             | 1             |
| 26  | 11010                              | 000                       | x            | 1             | 1             | 0                                | 1             | 0             |
| 27  | 11011                              | 000                       | x            | 1             | 1             | 0                                | 1             | 1             |
| 28  | 11100                              | 000                       | x            | 1             | 1             | 1                                | 0             | 0             |
| 29  | 11101                              | 000                       | x            | 1             | 1             | 1                                | 0             | 1             |
| 30  | 11110                              | 000                       | x            | 1             | 1             | 1                                | 1             | 0             |

Table continues on the next page...

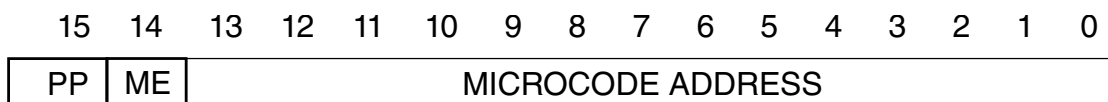
**Table 35-334. Alternate channel condition encoding scheme (continued)**

| No. | Encoded channel conditions [C4-C0] | Host Service Request bits | Link request | MatchA/TransB | MatchB/TransA | In/Output pin state <sup>1</sup> | Channel Flag1 | Channel Flag0 |
|-----|------------------------------------|---------------------------|--------------|---------------|---------------|----------------------------------|---------------|---------------|
| 31  | 11111                              | 000                       | x            | 1             | 1             | 1                                | 1             | 1             |
|     |                                    | Host Service Request      |              |               |               |                                  |               |               |

1. The ETPD field in [CCR\\_ENG1](#) and [CCR\\_ENG2](#) selects between input and output pin state.

### 35.15.1.1.5 Entry Point format

Entry Point information includes a Preload-Parameter selection field, a Match Enable field, and the first microcode address of the thread. The Entry Point format is illustrated in the following figure.

**Figure 35-293. Entry Point format**

#### PP - Preload Parameter

PP indicates which pair of channel parameters are loaded into registers P and DIOB from the SDM prior to the execution of a thread. Preloading occurs during the time-slot transition period (see [Time Slot Transition](#)).

1 = microengine register P is preloaded from parameter 2 and DIOB from parameter 3.

0 = microengine register P is preloaded from parameter 0 and DIOB from parameter 1.

The parameter numbers are offsets from the channel parameter base address. For more info, see [Parameter Access](#).

#### ME - Match Enable

ME specifies whether match event recognitions are enabled or disabled for the thread associated with the entry point during the thread execution. If they are disabled, a match recognition can only occur after channel service. For more details refer to [Match Recognition](#).

1 = Matches are enabled during the thread.

0 = Matches are disabled during the thread.

The disabling of Match A/B recognition by MEF is dependent on IPACA/B configuration on the serviced channel (see [Pin Control Registers](#)). If IPACA=1xx, Match A is not disabled by ME=0. Likewise, IPACB=1xx overrides the effect of ME on Match B to "always on". If IPACA/B=0xx, Match A/B is disabled for two microcycles during TST (see [Time Slot Transition](#)) and is re-enabled when Entry Point is loaded, if ME=1. Note that if the comparator is in equal-only mode and the time base reaches the value of the Match register during the time that recognition is disabled (beginning of TST, plus whole thread if ME=0), the match recognition is lost. If the comparator is in greater-equal mode, the match event may be recognized after the disabling period if it satisfies the "greater-than" condition.

#### MICROCODE ADDRESS - Starting Microcode Address

This field specifies the microcode address on which the thread is to begin execution.

### 35.15.1.2 Time Slot Transition

The Time Slot Transition period (also called TST for short) is the interval between the servicing of two channels, during which all Channel-specific context is loaded for the new serviced Channel. The primary tasks completed during this period include:

- Set MEF for the TST wait-states, if any.
- Reset MEF for the first two microcycles (after eventual wait-states).
- Update of the CHAN register with the number of the new channel to be serviced.
- Parallel update of ERTA and ERTB from CaptureA and CaptureB registers of the new serviced channel.
- Sampling of the branch conditions of the new channel to be serviced into the branch logic (this means flags TDLA/B, MRLA/B, LSR, FM[1], FM[0], and PSS). The branch conditions are coherent with the timebase capture values sampled into ERTA/B (if MRLA/B, TDLA/B are set at the same time of the sampling, either both old flag state and capture values are sampled, or both new values are sampled).
- Formation of the entry point address.
- Copy the ME bit in the Entry Point into MEF.
- Access to the entry point location and getting the first microinstruction address.
- Preload of two parameters from the SDM into P (32 bits) and DIOB (24 bits).
- Fetch the first instruction of the thread to be executed for the new channel.
- Preset the RAR register value (see [RAR Register](#)).

The preload operation is 32-bit wide for P and 24-bit wide for DIOB. The P register is loaded with all the 32-bit parameter. The DIOB register is loaded with the lower 24-bits of the parameter. The microcode can switch at any time to access the lower 24-bits, upper byte, or all the 32-bits of any parameter in the SDM. Preload of P-DIOB pair of parameters is atomic with respect to Host and CDC accesses, and so are coherent with their dual-parameter coherent transfers. For more details see [Parameter sharing and coherency](#).

No instructions are executed at the Engine where the time slot transition period occurs, but the other Engine can execute normally. Match A/B is unconditionally disabled on the first two TST microcycles, if IPACA/B=0xx (respectively). During the rest of time slot transition, match recognition can be disabled or not, depending on IPACA/B field and ME. See [Match Recognition](#).

Time Slot Transition takes a minimum of 3 microcycles (6 eTPU clocks), which may be extended due to SDM arbitration wait-states for the first preload access (see [SDM arbitration](#)). When no wait-states are received ([Figure 35-294](#)), DIOB is preloaded twice, one for each PP value, and the correct value remains in DIOB when the Entry Point is loaded. [Figure 35-295](#) and [Figure 35-296](#) show the timing for one and two wait-states, respectively.

Registers B, C, D and SR are not altered by TST and keep their values from the previous thread. The values of registers A, MACL and MACH are not guaranteed at the thread start.

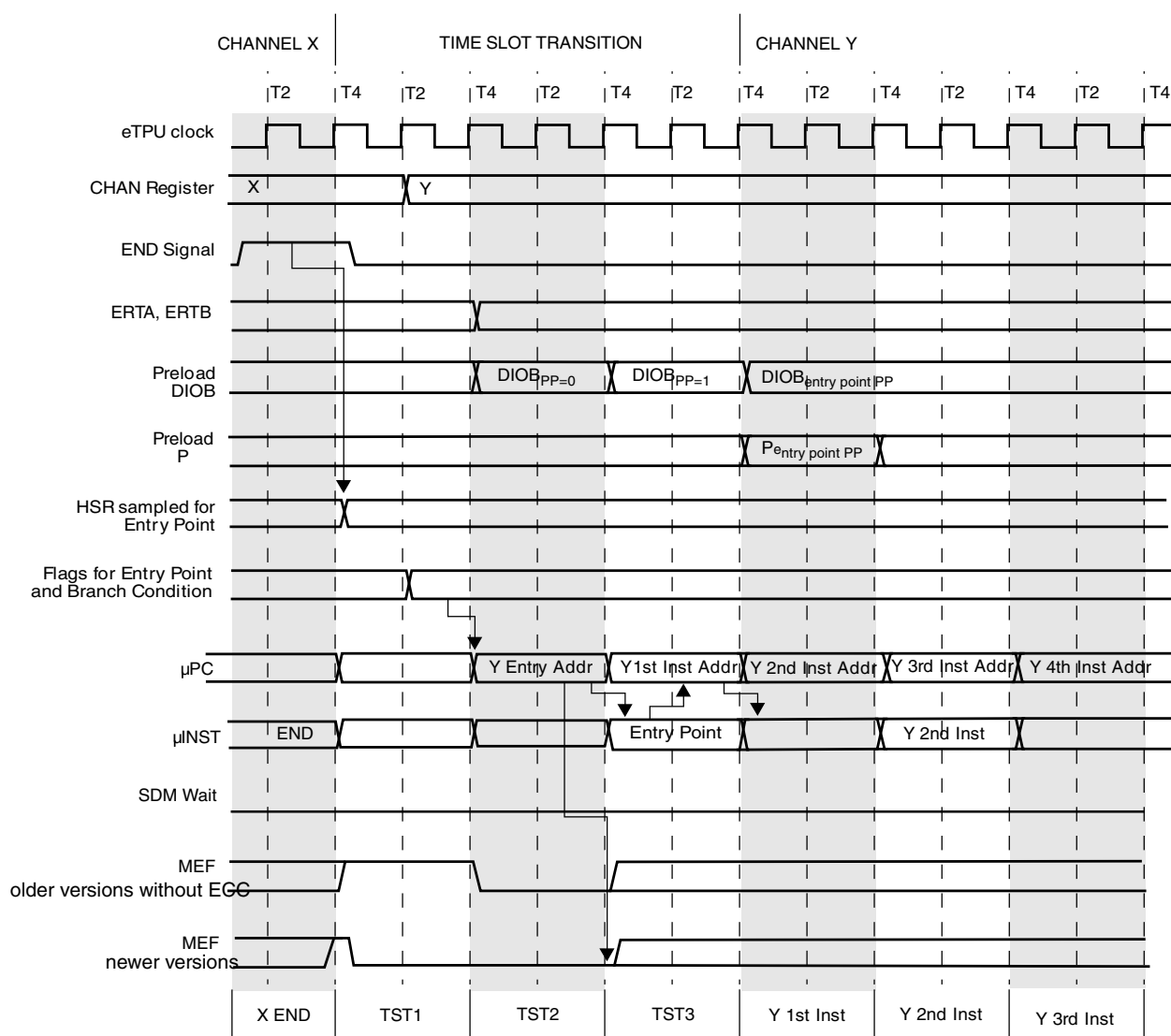


Figure 35-294. TST Timing - No Wait-states

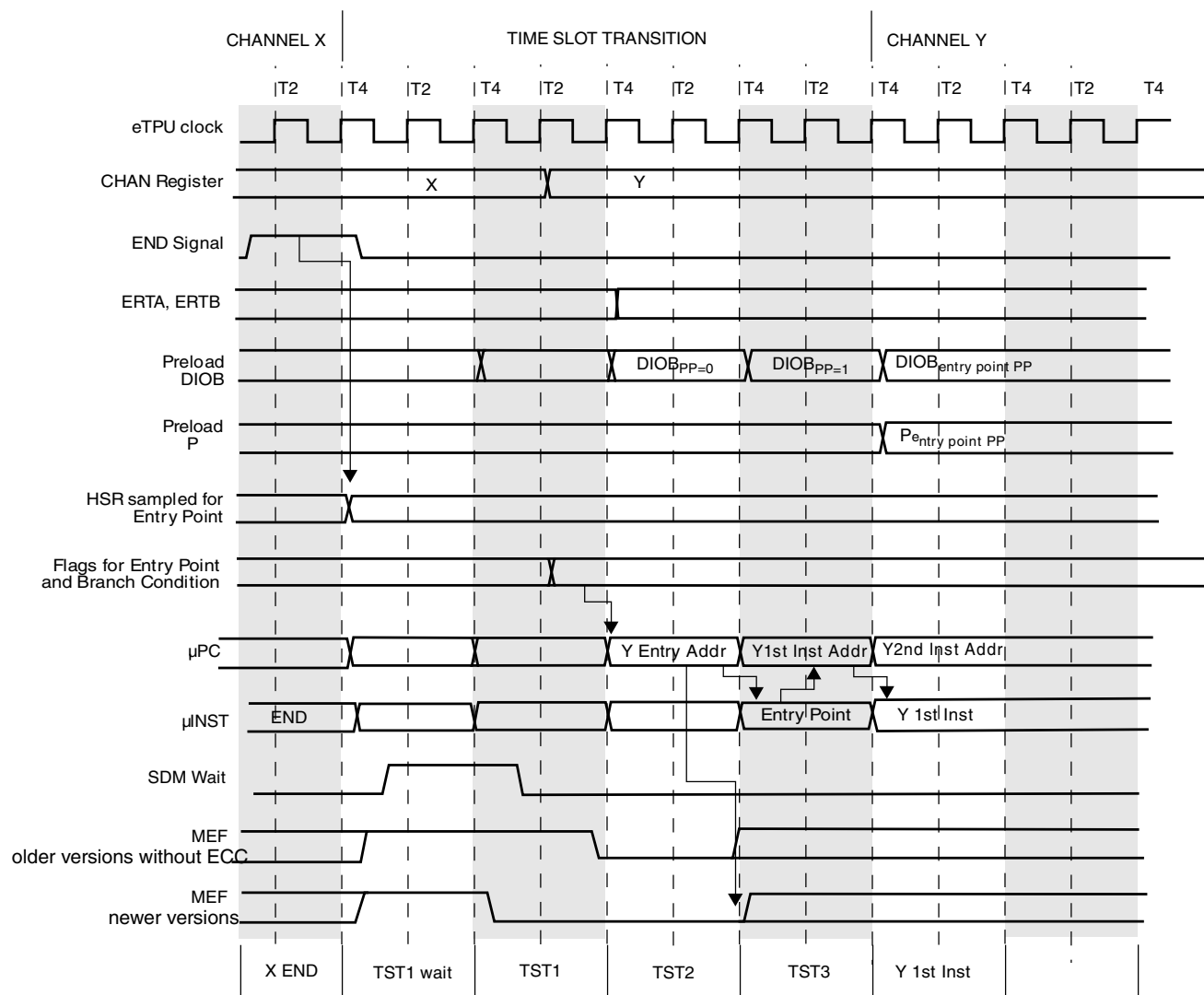


Figure 35-295. TST Timing - 1 Wait-State



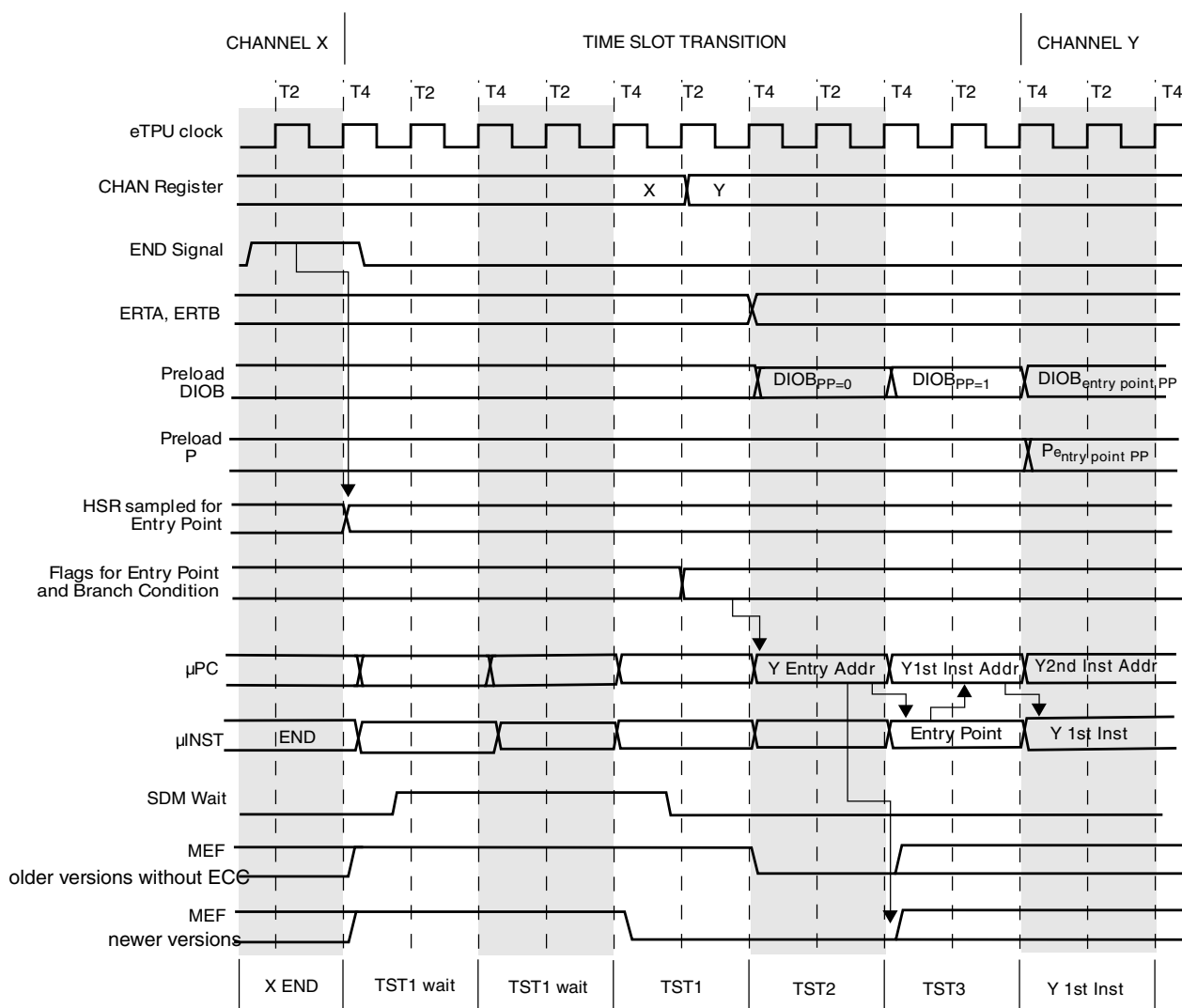


Figure 35-296. TST Timing - 2 Wait-states

For more information on Channel-specific registers and flags, refer to [Enhanced Channels](#). For more information on P, ERTA/B and DIOB registers refer to [Registers](#).

### 35.15.1.2.1 Thread gap

The MEF timing has changed from previous eTPU/eTPU2 versions, now being reset for two microcycles from the start of TST.<sup>12</sup> Therefore, two consecutive threads with ME=0 serving the same channel could block any match until the end of the second thread (see [Figure 35-294](#)). This could cause latency problems and also break compatibility with legacy code, so the thread transition logic was modified to introduce a one-microcycle gap between consecutive threads servicing the same channel, as shown in the following

12. the new MEF timing is found on all eTPU2 instances starting from the first version with Memory Error Support; all previous eTPU2 versions maintain the former MEF timing (see [Legacy Time Slot Transition Timing](#)).

figure. During the gap MEF is set to one, regardless of the previous or the next entry point ME field. The thread gap is inserted only between threads serving the same channel.

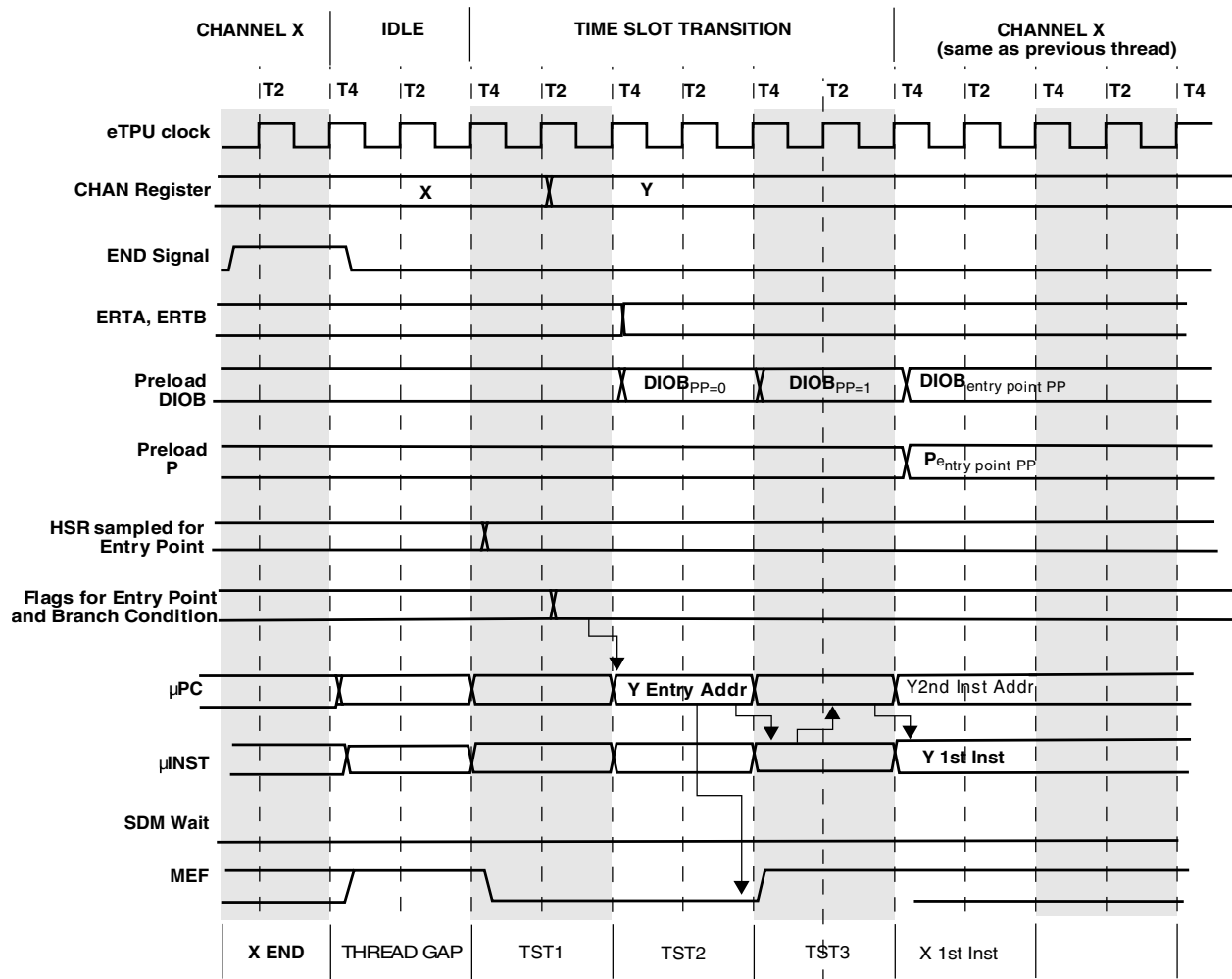


Figure 35-297. Thread gap

### 35.15.1.3 Thread ending

Threads can finish by either:

- an instruction with the END field active (see [Ending Current Thread - END](#)).
- a forced END by host writing to the FEND field in [eTPU Engine 1 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG1\)](#) and [eTPU Engine 2 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG2\)](#).

- a forced END caused by Watchdog timeout (see [Watchdog](#)).
- a forced END caused by a non-correctable memory error (see [Memory error support](#)).

### 35.15.1.4 Watchdog

Each engine has a watchdog mechanism to prevent a thread or a sequence of threads from running too long, impacting the latency of the other channel services. The watchdog is configured through [eTPU Watchdog Timer Register for Engine 1](#) (eTPU\_ER\_WDTR\_ENG1) and [eTPU Watchdog Timer Register for Engine 2](#) (eTPU\_ER\_WDTR\_ENG2). When the watchdog is enabled, an internal counter increments on each microcycle when a thread is executing. If the count is greater than the value specified in the WDCNT field in [eTPU Watchdog Timer Register for Engine 1](#) (eTPU\_ER\_WDTR\_ENG1) and [eTPU Watchdog Timer Register for Engine 2](#) (eTPU\_ER\_WDTR\_ENG2) and a thread is still executing, the watchdog:

1. If the WMO field in [eTPU Watchdog Timer Register for Engine 1](#) (eTPU\_ER\_WDTR\_ENG1) and [eTPU Watchdog Timer Register for Engine 2](#) (eTPU\_ER\_WDTR\_ENG2) is negated, forces an END of the thread, allowing other channels to be serviced
2. Sets the WDS status bit of the serviced channel in [eTPU Watchdog Status Register for Engine 1](#) (eTPU\_GCR\_WDSR\_ENG1) and [eTPU Watchdog Status Register for Engine 2](#) (eTPU\_GCR\_WDSR\_ENG2).
3. If the WMO field in [eTPU Watchdog Timer Register for Engine 1](#) (eTPU\_ER\_WDTR\_ENG1) and [eTPU Watchdog Timer Register for Engine 2](#) (eTPU\_ER\_WDTR\_ENG2) is negated, the channel is disabled, not initiating any thread until its WDS bit is cleared (however, its CPR field in [eTPU Channel n Configuration Register for Engine 1](#) (eTPU\_CCCR\_CnCR\_ENG1) and [eTPU Channel n Configuration Register for Engine 2](#) (eTPU\_CCCR\_CnCR\_ENG2) is not changed).
4. Issues a Global Exception and sets [WDTO1](#) (engine 1) or [WDTO2](#) (engine 2), regardless of the WMO bit state.

If the WMO field in [eTPU Watchdog Timer Register for Engine 1](#) (eTPU\_ER\_WDTR\_ENG1) and [eTPU Watchdog Timer Register for Engine 2](#) (eTPU\_ER\_WDTR\_ENG2) is asserted, the watchdog can be used to monitor the execution time threshold of a thread or thread sequence without interfering with the eTPU code execution.

The watchdog can be configured in one of the following modes, defining how the internal watchdog count is reset:

- **Thread Length Mode:** the watchdog count is reset at the end of each thread.
- **Busy Length Mode:** the watchdog count is reset when the microengine goes idle. A sequence of threads, one right after another, keeps the count running. The counter is also reinitialized when a thread is forced to end, so that a new count begins if another TST initiates at the following microcycle.

The following applies to the watchdog mechanism:

- microcycles during TST and SDM access wait-states (on TST or instruction execution) are counted.
- if the watchdog count equals WDCNT in the last microinstruction (with SDM wait-states or not) of a thread servicing a channel, its WDS bit is not set.
- if the watchdog count expires (gets greater than WDCNT) during the TST, the thread is forced end on its first instruction.
- the watchdog count does not wrap, so that a thread (in thread length mode) or a thread sequence (in busy length mode) that lasts for more than the maximum value of WDCNT does get a forced end.

### **Note**

Watchdog must not be enabled when the microengine enters halt mode.

The counter does not run when the engine is stopped, and resets when the watchdog is disabled.

## **35.15.2 Host interface**

### **35.15.2.1 System configuration**

System Configuration Registers are described in [System Configuration memory map and registers](#). Detailed explanation of the configured functionalities is found throughout [Functional description](#) and a specification for the initial configuration sequence is found in [Configuration sequence](#).

## 35.15.2.2 Interrupts and data transfer requests

### 35.15.2.2.1 Interrupt types and sources

Each one of the eTPU channels can be a source of two requests: **Channel Interrupt** request and **Data Transfer Request**. Channel Interrupts are targeted to a Host CPU. Data Transfer Requests may be targeted to a data transfer module (for example, a DMA controller). Interrupt and Data Transfer registers are used by the Host to enable interrupts and data transfer requests, indicate their status and service them. Interrupt and Data Transfer requests have the same sets of registers and external signals, and are handled in the same way. They differ only by the fact that Data Transfer Requests are also cleared by the assertion of respective DMA completion acknowledge line.

#### Note

Interrupt and Data Transfer requests can be cleared even when Engines are in Module Disable Mode, through the Global Channel Registers, and also DMA completion for Data Transfer requests.

Channel Interrupts and Data Transfer Requests can only be issued by eTPU microcode, through one of the Channel Control instruction fields (see [Channel Interrupt and Data Transfer Requests](#)).

Both Channel Interrupt and Data Transfer requests can be individually enabled for each channel.

eTPU Interrupt and Data Transfer Registers are mirrored in two organizations: grouped by Channel and grouped by type (interrupt status, interrupt enable, data transfer status, data transfer enable). This allows either "channel-oriented" or "bundled channel" Host interrupt service schemes, or a combination of them. For a detailed description, refer to the Channel Register Layout and Global Channel Registers sections.

eTPU can also assert a **Global Exception** interrupt indicating a global illegal state. There are six possible sources for a Global Exception:

- Execution of an illegal instruction by the microengine (see [Illegal instructions](#)). This Global Exception source is flagged by [ILF1](#) and [ILF2](#).
- An SCM signature mismatch detected by the Multiple Input Signature Calculator - MISC. See [SCM Test - Multiple Input Signature Calculator](#). This source is flagged by [SCMMISF](#).

- Microcode request, through microinstruction field CIRC (see [Channel Interrupt and Data Transfer Requests](#)). This Global Exception source is flagged by [MGE1](#) (Engine 1) and [MGE2](#) (Engine 2). The cause of this illegal state is application-dependent. The microcode may write an error code into the SDM to indicate the cause of the exception, for instance.
- A Watchdog timeout, flagged by [WDTO1](#) and [WDTO2](#) (see [Watchdog](#)).
- An SDM or SCM non-correctable error on a microengine or host access, flagged by [SDMERR](#) and [SCMERR](#), respectively (see [Memory error support](#)).
- Bus errors on microengine accesses to the Shared Subsystem Area, flagged by [SSAERR](#) (see [Shared Subsystems support](#)).
- The latency measured by any of either engine's Latency Monitors is greater than the specified maximum, flagged by [LTO1](#) (Engine 1) and [LTO2](#) (Engine 2) (see [Latency Monitor](#)).

Global Exceptions cannot be directly disabled within eTPU, except by disabling its sources (Memory Error, SSS, MISC and microcode), and it is cleared by writing 1 to [GEC](#). Clearing Global Exception clears all Global Exception source status bits (ILF1, ILF2, SCMMISF, MGE1, MGE2, SSAERR, SCMERR, SDMERR). If GEC is written 1 at the same time any of the sources issues a Global Exception, both the interrupt and the status bit of that source remains asserted. The assertion of Global Exception by one of the sources above does not prevent the others from asserting it too, so any number of them, in any combination, can be flagged.

### Note

There can be a race between the clear of a Global Exception and occurrence of a new set condition, such that the set happens just before the clear and cannot be sensed by the Host. Therefore, Global Exception cannot be used as a normal interrupt source. It should only be used for emergency procedures.

Memory Error Correction support (see [Memory error support](#)) has a distinct interrupt source, which can activate on any correctable error occurrence, either to SCM or SDM accesses, but does not activate on non-correctable errors (which assert Global Exception instead). Memory Error Correction interrupt assert when it is enabled through [MECIE](#) (see [eTPU Memory Error Status Register \(eTPU\\_MESR\\_MESR\)](#)) and either DCERR or CCERR in [eTPU Memory Error Status Register \(eTPU\\_MESR\\_MESR\)](#) is asserted. The interrupt is negated when both DCERR and CCERR are negated, or MECIE is negated.

### 35.15.2.2.2 Interrupt and Data Transfer Request overflow

If a Channel Interrupt was issued, its status bit is still set, and microcode issues another Channel Interrupt, the Interrupt Overflow status bit is set for that channel. Interrupt Overflow status can be checked by the Host in the CIOS field of [eTPU Channel n Status Control Register for Engine 1 \(eTPU\\_CCCR\\_CnSCR\\_ENG1\)](#) and [eTPU Channel n Status Control Register for Engine 2 \(eTPU\\_CCCR\\_CnSCR\\_ENG2\)](#), mirrored in [eTPU Channel Interrupt Overflow Status Register for Engine 1 \(eTPU\\_GCR\\_CIOSR\\_ENG1\)](#) and [eTPU Channel Interrupt Overflow Status Register for Engine 2 \(eTPU\\_GCR\\_CIOSR\\_ENG2\)](#). Interrupt Overflow status is not cleared automatically when Interrupt Status is cleared. The same mechanism and respective registers ([eTPU Channel Data Transfer Request Overflow Status Register for Engine 1 \(eTPU\\_GCR\\_CDTROSR\\_ENG1\)](#) and [eTPU Channel Data Transfer Request Overflow Status Register for Engine 2 \(eTPU\\_GCR\\_CDTROSR\\_ENG2\)](#)) are available for Data Transfer Requests.

If interrupt is set and cleared at the same time, set prevails and overflow is not altered (keeps the same state as it was before, asserted or not).

Global Exception has no overflow status.

### 35.15.2.3 Parameter Access

#### 35.15.2.3.1 Parameter Access Widths

From the Host side the SDM address space is mapped in bytes, and each 32-bit parameter occupies 4 contiguous, aligned bytes. The Host can read/write the SDM by 8-, 16-, or 32-bit accesses in aligned addresses.

In 32-bit access, Host can access all 32 bits or only the lower 24 bits with an automatic sign extension (see [Parameter Sign Extension area](#)).

#### 35.15.2.3.2 Parameter Addresses and Endianness

To access parameter number xxx, eTPU Microengine(s) would select address xxx. The Host would add  $(xxx*4)$  to the SDM base address to access the same parameter. For example, parameter 0x101 is seen by the Host in  $(SDM\ base\ address + 0x404)$ . An example of SDM memory map is shown in [Figure 35-298](#). The Host can access the SDM with a 32-bit-wide bus cycle to a four-byte aligned address, 16-bit-wide bus cycle to a two-byte aligned address, or 8-bit wide bus cycle to any byte address.



The address of the 24-bit parameters and the most significant byte depends on the endianness of the MCU. For more details, see the [Endianness](#).

### 35.15.2.3.3 Parameter concurrency

Host accesses to parameters may occur in parallel with eTPU Microengine accesses. Readings taken from a group of parameters while they are being simultaneously updated may lack coherency. eTPU provides mechanisms to ensure parameter coherency in accesses from both Host side and Microengine side, including the use of a coherent dual-parameter transfer mechanism, described in detail on [Parameter sharing and coherency](#).

### 35.15.2.3.4 Parameter Sign Extension area

The SDM address space to the Host is mirrored in a Parameter Sign Extension (PSE) area (see [Table 35-4](#)). Accesses from the Host to the PSE area differ from accesses to the standard SDM address space as follows:

- **Writes:** the most significant byte of the parameters is not written, and the SDM retains the old byte value, regardless of the Host access size.
- **Reads:** the most significant bit of the 24-bit parameter (that is, the MSB of the second most significant 32-bit parameter byte) is repeated in the 8 most significant bits of the read value on all 32-bit reads and most significant 16- and 8-bit reads.

The same parameters written in the standard SDM address space are read from the PSE area with the same offsets, and vice-versa. See [Table 35-429](#) for a reference of the address offsets in big and little endian machines.

This feature relieves the Host from extending the signal of 24-bit eTPU parameters before calculations, and from read-modify-write accesses to modify 24-bit parameters at the SDM.

### 35.15.2.4 SDM organization

The SDM internal partition for channel allocation is dynamic and programmed in the Channel Registers (see [eTPU Channel n Configuration Register for Engine 1 \(eTPU\\_CCCR\\_CnCR\\_ENG1\)](#) and [eTPU Channel n Configuration Register for Engine 2 \(eTPU\\_CCCR\\_CnCR\\_ENG2\)](#)).



The Host application is responsible for allocating a different parameter base address to each channel during the initial eTPU configuration, and to allocate enough parameters for the selected Function, with no unintentional overlapping between parameters of different functions.

Besides channel parameters, global areas may have to be allocated for parameters that are shared by more than one channel, in one or both Engines. Also, temporary parameter areas should be reserved to be used by the coherent parameter transfer mechanisms described in [Parameter sharing and coherency](#), if necessary. An SDM organization example is shown in the following figure.

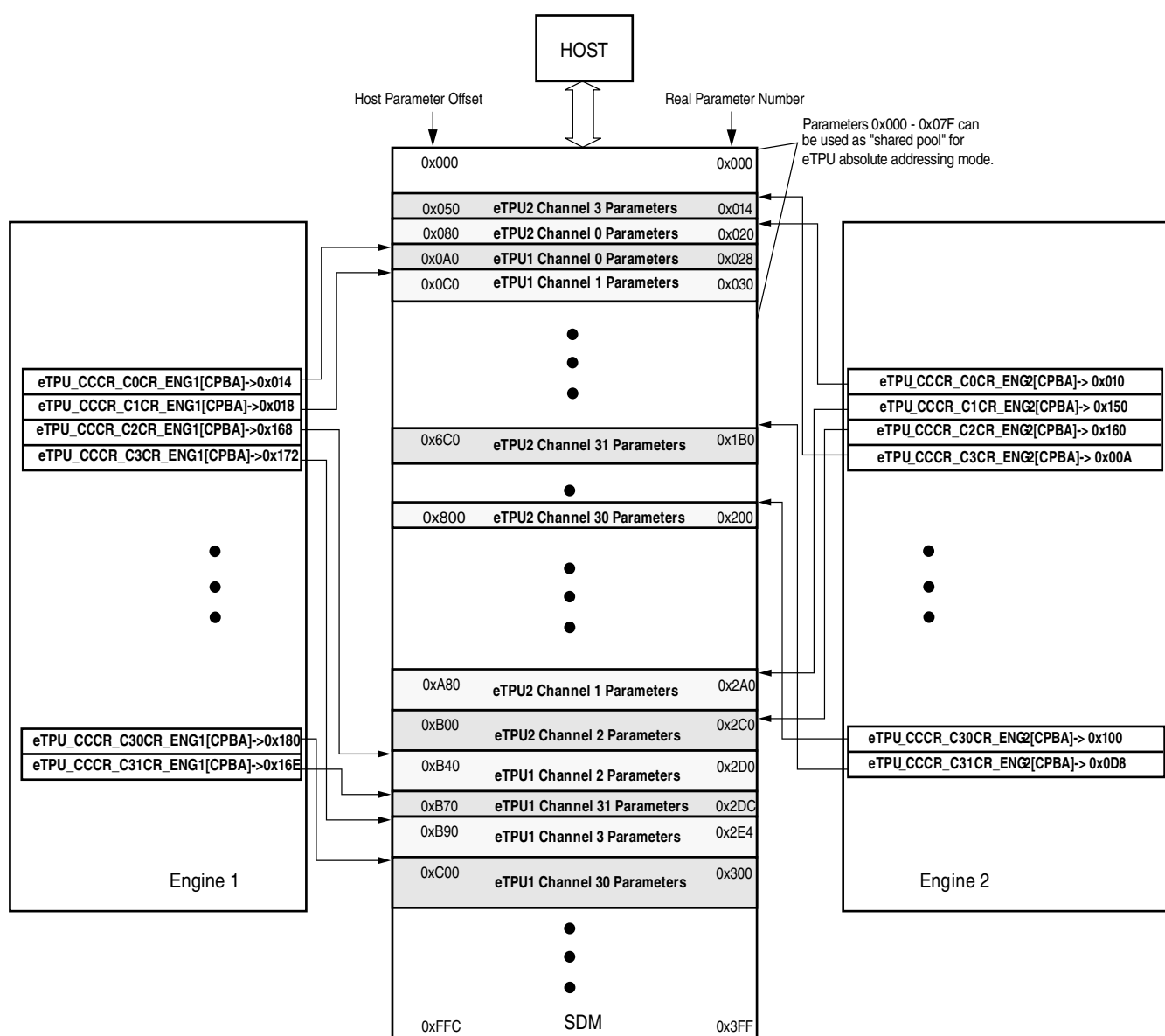


Figure 35-298. SDM Organization Example (4 KB)

Single-engine eTPU or dual eTPU system may require less parameters than the maximum number provided by the SDM. Since the SDM partition is fully dynamic, there is no limitation of fixed channel addresses, and the reduced array can be fully utilized.

### 35.15.2.5 Host Service Requests

Host CPU can request immediate service from a channel by writing a non-zero value to the Host Service Request register field - HSR (see [eTPU Channel n Host Service Request Register for Engine 1 \(eTPU\\_CCCR\\_CnHSRR\\_ENG1\)](#) and [eTPU Channel n Host Service Request Register for Engine 2 \(eTPU\\_CCCR\\_CnHSRR\\_ENG2\)](#)). There is one HSR field for each channel, so that writing to it generates a Service Request to the respective channel only. A zero value in HSR means no Host Service Request is pending for the channel.

HSR value turns to 000 automatically at the end of microengine service for that channel, but only if the thread started due to an HSR.

The meaning of a non-zero HSR value depends on the Function assigned for the channel. These bits are part of the conditions which select the Function entry point, and cannot be tested by microcode. For more details, refer to [Entry Points](#).

If Host writes HSR=000 when a thread for the same channel is already running, the thread runs until the end and is not aborted. If Host writes HSR>000 when an HSR thread for the same channel is already running, HSR value resets at the end of the thread, and no new HSR will be pending. If HSR is written before its value is resolved by the scheduler during TST, the entry point will obey the new HSR value, and if this new value is 000, no service thread is executed for the HSR.

The scheduling of HSRs is completely asynchronous with Host accesses, and there is no race-free manner to change an HSR value before service thread execution, so generally the safe way is: write HSR>0 only when HSR=0. Error recovery or emergency host procedures may require one to safely abort service and reset channel state when an HSR is already pending or executing. In these cases, the procedure below should be followed:

1. Disable the channel, writing CPR=00 in [eTPU Channel n Configuration Register for Engine 1 \(eTPU\\_CCCR\\_CnCR\\_ENG1\)](#) and [eTPU Channel n Configuration Register for Engine 2 \(eTPU\\_CCCR\\_CnCR\\_ENG2\)](#). That will prevent any pending HSR to be serviced.
2. Check if the channel is currently being serviced, reading its service status bit in [eTPU Channel Service Status Register for Engine 1 \(eTPU\\_GCR\\_CSSR\\_ENG1\)](#) and [eTPU Channel Service Status Register for Engine 2 \(eTPU\\_GCR\\_CSSR\\_ENG2\)](#). If

it is, wait for the time necessary to finish the service pending, or check again until  $HSR == 0$ , or channel service bit in [eTPU Channel Service Status Register for Engine 1 \(eTPU\\_GCR\\_CSSR\\_ENG1\)](#) and [eTPU Channel Service Status Register for Engine 2 \(eTPU\\_GCR\\_CSSR\\_ENG2\)](#) is cleared.

3. Write HSR with the error recover value. This value should, possibly combined with other host-defined flags in SDM or FM bits, initiate a channel reset or error recovery procedure.
4. Re-enable the channel, writing CPR value  $> 0$  in [eTPU Channel n Configuration Register for Engine 1 \(eTPU\\_CCCR\\_CnCR\\_ENG1\)](#) and [eTPU Channel n Configuration Register for Engine 2 \(eTPU\\_CCCR\\_CnCR\\_ENG2\)](#).

### 35.15.2.6 SCM access

Only Host can access SCM as data. Depending on the specific device, SCM may be implemented as a RAM or ROM. This determines Host accesses to the SCM as shown below.

#### 35.15.2.6.1 SCM RAM implementations

When SCM is implemented as RAM, the Host may read or write to SCM by setting [VIS](#) = 1. If  $VIS=0$  and Host tries to access SCM space, a bus error is issued, writes are ineffective and read data is meaningless. Both Engines must be stopped or halted to set  $VIS=1$ .

#### Note

It is necessary to turn [VIS](#) bit on to set software breakpoints (see [Software breakpoints](#)).

#### 35.15.2.6.2 SCM low power

SCM turns off its internal clocks when both Engines are stopped (STF field in [eTPU Engine 1 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG1\)](#) and [eTPU Engine 2 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG2\)](#) asserted), [VIS](#), and [MISC](#) is not enabled ( $SCMMISEN=0$ ). The SCM clocks are automatically turned on if either one of the STF bits is negated or [VIS](#) turns to 1, or [SCMMISEN](#) turns to 1.

SCM clocks are not turned off if any of the Engines is not stopped, even if they are both halted.

The conditions for SCM Clocks and MISC activation are summarized in the following table.

**Table 35-335. SCM clocks and MISC activation**

| STF in<br>eTPU_SCR_ECR_<br>ENG1 | STF in<br>eTPU_SCR_ECR_<br>ENG2 | VIS in<br>eTPU_SCR_MCR | SCMMISEN in<br>eTPU_SCR_MCR | SCM clocks | MISC             |
|---------------------------------|---------------------------------|------------------------|-----------------------------|------------|------------------|
| 0                               | x                               | 0 <sup>1</sup>         | 1                           | On         | On               |
| 0                               | x                               | 0 <sup>1</sup>         | 0                           | On         | off              |
| x                               | 0                               | 0 <sup>1</sup>         | 1                           | On         | On               |
| x                               | 0                               | 0 <sup>1</sup>         | 0                           | On         | off              |
| 1                               | 1                               | 0                      | 0                           | off        | off              |
| 1                               | 1                               | 0                      | 1                           | On         | On               |
| 1 <sup>2</sup>                  | 1 <sup>2</sup>                  | 1                      | 0                           | On         | off              |
| 1 <sup>2</sup>                  | 1 <sup>2</sup>                  | 1                      | 1                           | On         | off <sup>3</sup> |
| 0                               | 0                               | x                      | 0                           | On         | off              |
| 0                               | 0                               | x                      | 1                           | On         | On               |

1. VIS cannot be written 1 if eTPU\_SCR\_ECR\_ENG1[STF]=0 or eTPU\_SCR\_ECR\_ENG2[STF]=0, and both HLTF bits are 0.
2. If VIS=1, neither MDIS can be written 0 nor the Engine leaves Stop Mode, regardless of device stop request.
3. MISC resets and stays so when VIS=1, restarting automatically when VIS goes 0 if SCMMISEN=1.

### 35.15.2.6.3 SCM Off-range data

When read accesses are made, either by the Host or by a microengine, to addresses above the limit corresponding to [SCMSIZE](#), the value read comes from [eTPU SCM Off-range Data Register \(eTPU\\_SCR\\_SCMOFFDATAR\)](#). The Host can program the register at initialization with an opcode value with operations that try to protect or recover the system from runaway code, for instance: terminate the thread, clear channel flags, disable match and transition service requests, issue an interrupt, jump to an error recovery procedure<sup>13</sup>. Writes to unimplemented addresses do not return error and can write on unspecified mirror addresses, so they should be avoided.

### 35.15.2.7 Bus error conditions

Follows a summary of the possible causes of bus errors on host accesses to the eTPU memory space:

- Read or write access to the SCM with [VIS](#) =0.

13. Only part of these suggested operations can be parallelized in a single instruction, see [Microinstruction formats](#).

- Read or write access to the channel registers with eTPU\_SCR\_ECR\_ENG $n$  bits MDIS=1, or bit STF=1 (see [eTPU Engine 1 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG1\)](#) and [eTPU Engine 2 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG2\)](#)).
- Write access to the Time Base Registers with eTPU\_SCR\_ECR\_ENG $n$  bits MDIS=1, or bit STF=1 (see [eTPU Engine 1 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG1\)](#) and [eTPU Engine 2 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG2\)](#)).
- Non-correctable errors in the SCM on reads of any size or writes less than 32-bits wide when CEDD =0.
- Non-correctable errors in the SDM on reads when DEDD =0.
- Accesses to the Shared Subsystem Area, when propagating a bus error issued by the attached subsystem.

### 35.15.3 Scheduler

Every Function is composed of one or more Threads. A Thread consists of a group of instructions that, once begins execution, cannot be interrupted by host or channel events. Each active channel intends to be serviced, being granted time for Thread execution. Since one microengine handles several channels operating concurrently, the Function threads must be executed serially.

The task of the Scheduler is to recognize and prioritize the channels needing service and to grant execution time to each channel. The time given to an individual Thread for execution or service is called a **Time Slot**. The duration of a time slot is determined by the number of instructions executed in the Thread plus SDM wait-states received, and varies in length.

At any time, an arbitrary number of channels can require service. To request service, channel logic, eTPU microcode or Host application notifies the Scheduler by issuing a Service Request.

#### 35.15.3.1 Channel enabling and priority assignment

Every channel is assigned one of three priority levels - high, middle, or low - by the Host CPU, through the Channel Configuration Register field CPR (see [eTPU Channel  \$n\$  Configuration Register for Engine 1 \(eTPU\\_CCCR\\_C \$n\$ CR\\_ENG1\)](#) and [eTPU Channel  \$n\$](#)

[Configuration Register for Engine 2 \(eTPU\\_CCCR\\_CnCR\\_ENG2\)](#)). These registers are also used to disable the channel, which is equivalent to assigning it a "null" priority. In this case, the Scheduler does not grant any of its Service Requests.

It is possible to change the channel priority level or disable it dynamically. If the Host disables a channel when it is currently being serviced, channel service thread will complete. This means that it is possible for the output level of a channel signal to change, or a Host interrupt occur, even after its priority register was written to "null". For instance, if an output transition is scheduled, the transition will occur even after the channel is disabled.

Service requests previously pending or that occur while a channel is disabled remain asserted while the channel is disabled, and are serviced if the channel is enabled again, in due time determined by the priority scheme and concurrent requests from other channels. Channels are disabled after reset, and it is recommended to configure a Host Service Request for initialization of a channel before that channel is enabled to active priority (see [Initialization/application information](#)).

### 35.15.3.2 Channel priority schemes

The Scheduler holds a Service Grant register with one bit for each channel. Once the Scheduler grants a time slot to channel, the Service Grant bit for that channel is asserted in the Service Grant register. When the Service Grant bit of a channel is set, the channel may request new service but is not serviced again before its Service Grant bit is cleared.

When all channels in a same priority level are serviced, their Service Grant bits are cleared at the end of the thread, one eTPU clock before the next serviced channel is calculated, according to the scheme below<sup>14</sup>:

- Clear all grant bits of priority High if all channels of that priority that are requesting have their grant bits in 1.
- Clear all grant bits of priority Medium if all channels of that priority that are requesting have their grant bits in 1.
- Clear all grant bits of priority Low if all channels of that priority that are requesting have their grant bits in 1.
- Clear all grant bits of disabled channels.

---

14. Grant bits are also cleared in the next clock, when the service channel is chosen, or when the microengine is idle, using the same scheme.

This scheme assures that no channel is left with its grant bit forever asserted (preventing it from being serviced again), even if the channel priorities are reassigned during the execution.

Priority level is determined based on the maximum latency desired for each channel. A channel having a Function that requires the most frequent or more immediate service should be allocated a high priority level.

The eTPU employs a **primary** and a **secondary priority scheme**. These two schemes ensure frequent servicing of high-demand Functions and ensure a minimum time allocation to all channels requesting service, regardless of their priority level. The primary scheme prioritizes requesting channels that have different priority levels; the secondary scheme prioritizes requesting channels that have the same priority level.

Initially, a channel requests service and is granted a time slot by the Scheduler: Service Grant bit is asserted. If only high-level channels constantly receive service first because of their priority level, middle- and low-level channels would only be serviced by default, i.e., if no high-level channels request service. To ensure that each priority level receives an opportunity for servicing, every time slot has a fixed priority level that the Scheduler honors first. Divided into sets of seven, time slots are numbered from one to seven. The figure below illustrates the numbered time slots in sets of seven (fields A and B) and identifies their assigned default priority level. The high level has more time slots than the middle and low levels. Out of every seven time slots available, four are assigned to honor high-level channels first, two are assigned to honor middle-level channels first, and one is assigned to honor low-level channels first. Only one request (in each Engine) is serviced per time slot. When no channel requests service and the microengine is idle the priority scheme is initialized to time slot one, to prevent priority inversion on the next request<sup>15</sup>.

---

15. Priority inversion would occur in the following situation: no channel is requesting service, and the current time slot is primarily assigned to a low-priority channel. If the Scheduler was not reset to time slot one and two channels requested service at the same time, one with high priority and the other with low priority, the channel to be serviced would be the low-priority channel.

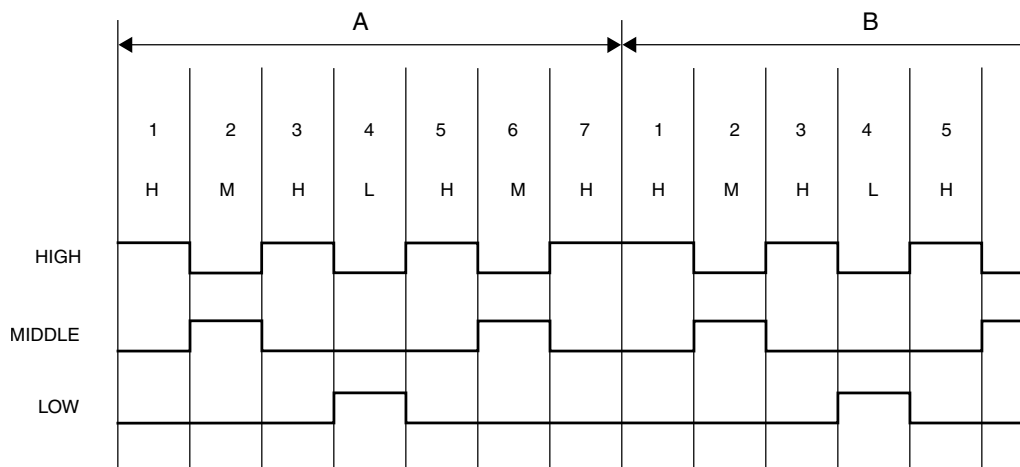


Figure 35-299. Time Slot Priority levels

### 35.15.3.2.1 Primary Scheme - Priority Among Channels on Different Levels

Although time slot priority assignment is fixed, the servicing priority is not. The primary scheme acknowledges the priority level assigned to a time slot, granting service first to a channel having the same priority. In [Channel priority schemes](#), time slot one has a high-level assignment; therefore, a high-level channel requesting service is recognized first. However, if no high-level channel requests service, the Scheduler recognizes a requesting middle-level channel. If this level has no request, the Scheduler continues to the low-level. If no requests occur, the Scheduler truncates the seven state cycle and starts a new cycle at time slot one, waiting for the first request. Granting service to a different-level channel is called priority passing. The order of passing always gives the highest priority to the assigned level, and the second priority to the higher of the remaining requesting priority levels as shown in the following table.

Table 35-336. Priority Passing

| Assigned Priority Level | Next Priority Level | Next Priority Level |
|-------------------------|---------------------|---------------------|
| High                    | Middle              | Low                 |
| Middle                  | High                | Low                 |
| Low                     | High                | Middle              |

When priority is passed to another level, that level is serviced and the fixed-priority-level sequence is resumed with the next time slot.



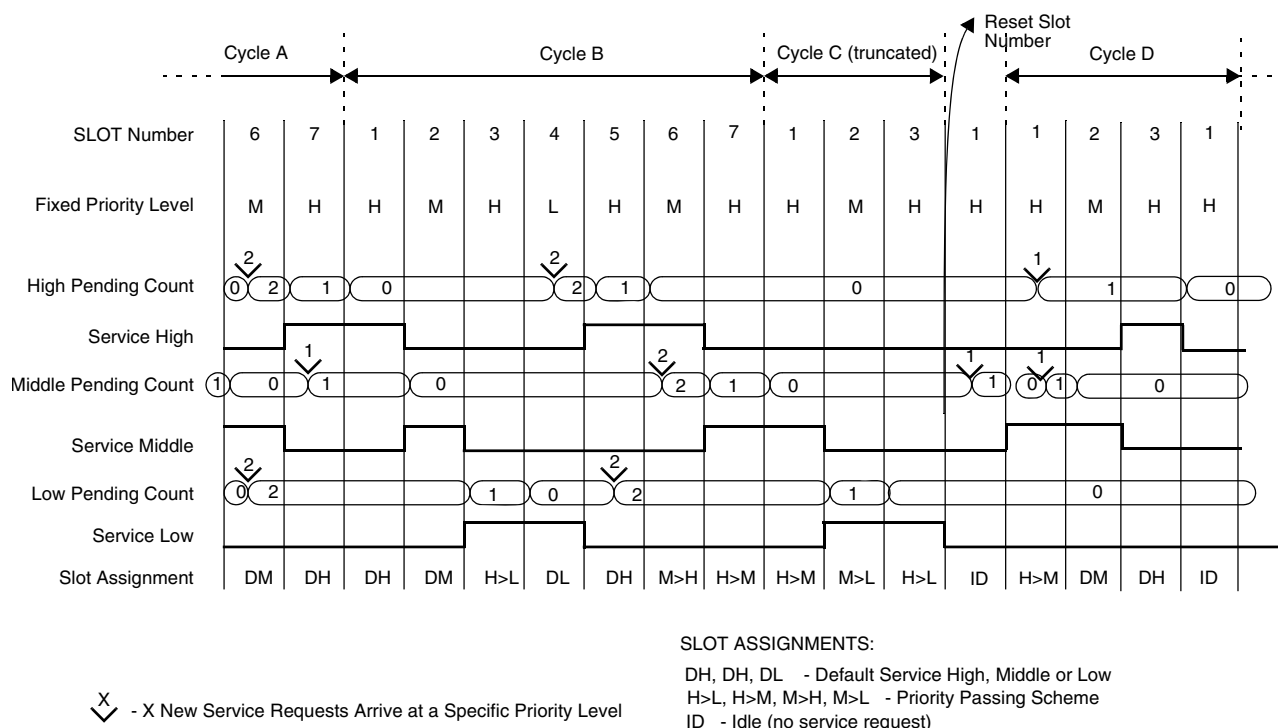


**Figure 35-300. Priority Passing Example**

Examples of priority passing are shown in the above figure. Each cycle contains seven time slots (or less if no service request exist). In cycle B, no high-level or middle-level service requests are present before time slot three which is assigned by default to high-level priority. Thus, time slot three is passed to the low level. In cycle B there are also no middle-level service requests before time slot six, so it passes the priority to a requesting high-level channel. During time slot six no more high level requests are left, but two new middle-level requests arrive, and there are also three low level pending service requests. Thus, time slot seven of cycle B and time slot one of cycle C are passed to the middle-level which is the next priority level after high. Time slots two and three of cycle C are passed to the low level which contains the three remaining channel service requests. At time slot three of cycle C the last low level request is serviced, and the Scheduler passes to idle state. At this point the cycle C is truncated and the Scheduler passes to time slot one of cycle D.

### 35.15.3.2.2 Priority passing disabling

The priority passing scheme allows a case where a high priority channel loses to a lower priority one right after another lower priority has been serviced, exemplified in the Cycle D on [Figure 35-300](#). A middle priority channel wins time slot 1 due to priority passing from high to middle. While it is being serviced, two new service requests arrive, one high and one middle priority. The high priority request loses to the middle one on next time slot 2 by default priority assignment.

This priority inversion can be avoided by setting the SPPDIS field in the eTPU\_SCR\_ECR\_ENG $n$  registers (see [eTPU Engine 1 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG1\)](#) and [eTPU Engine 2 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG2\)](#)), which disables the priority passing mechanism. When priority passing is disabled, at the end of the thread the slot number is incremented until a time slot that matches the priority of one of the requesting channel(s). The time slot advance takes no extra clocks. If no channel requests service, the time slot counter stays at time slot 1. The priority selection scheme with disabled priority passing is summarized in the following table.

**Table 35-337. Priority Passing Disabling**

| At the end of time slot | servicing priority | if any request of priority | service it on time slot | else if any request of priority | service it on time slot | else if any request of priority | service it on time slot |
|-------------------------|--------------------|----------------------------|-------------------------|---------------------------------|-------------------------|---------------------------------|-------------------------|
| 1                       | High               | Medium                     | 2                       | High                            | 3                       | Low                             | 4                       |
| 2                       | Medium             | High                       | 3                       | Low                             | 4                       | Medium                          | 6                       |
| 3                       | High               | Low                        | 4                       | High                            | 5                       | Medium                          | 6                       |
| 4                       | Low                | High                       | 5                       | Medium                          | 6                       | Low                             | 4                       |
| 5                       | High               | Medium                     | 6                       | High                            | 7                       | Low                             | 4                       |
| 6                       | Medium             | High                       | 7                       | Medium                          | 2                       | Low                             | 4                       |
| 7                       | High               | High                       | 1                       | Medium                          | 2                       | Low                             | 4                       |

An example of the priority passing disabling scheme is illustrated in the figure below. The sequence of service requests is the same as in the example of [Figure 35-300](#), and although the time slot incrementing differs, the priorities granted are the same for cycle B. Cycle C has one of the low priority channels serviced before the second middle one. Cycle D, however, no longer has the priority inversion.

In cycle B, after the time slot 2 only a low priority request remains, so the time slot count advances directly to 4, which has a low priority assigned. Time slot keeps on 4 for the next service, as only a low priority request remains also, and only time slot 4 is assigned to low. Two high priority services contend for the next time slot 5 (assigned to High). The second high priority channel is serviced on the next time slot, jumped to 7 because there is no middle request, ending cycle B. Cycle C starts with time slot 2, as there are no high priority requests and two middle and two low ones. After the first middle service, time slot count skips 3 assigned to high (no high requests), and services a low priority channel on time slot 4. It follows the same scheme until there are no other requests and cycle C is truncated, resetting the time slot counter to 1.

Cycle D begins with a middle request, jumping to time slot 2. During this service two requests arrive, one high and one middle. Unlike what happened with priority passing, the next serviced is the high priority channel, as the time slot increments to 3. The second middle priority channel request in cycle D is finally serviced next, on time slot 6.

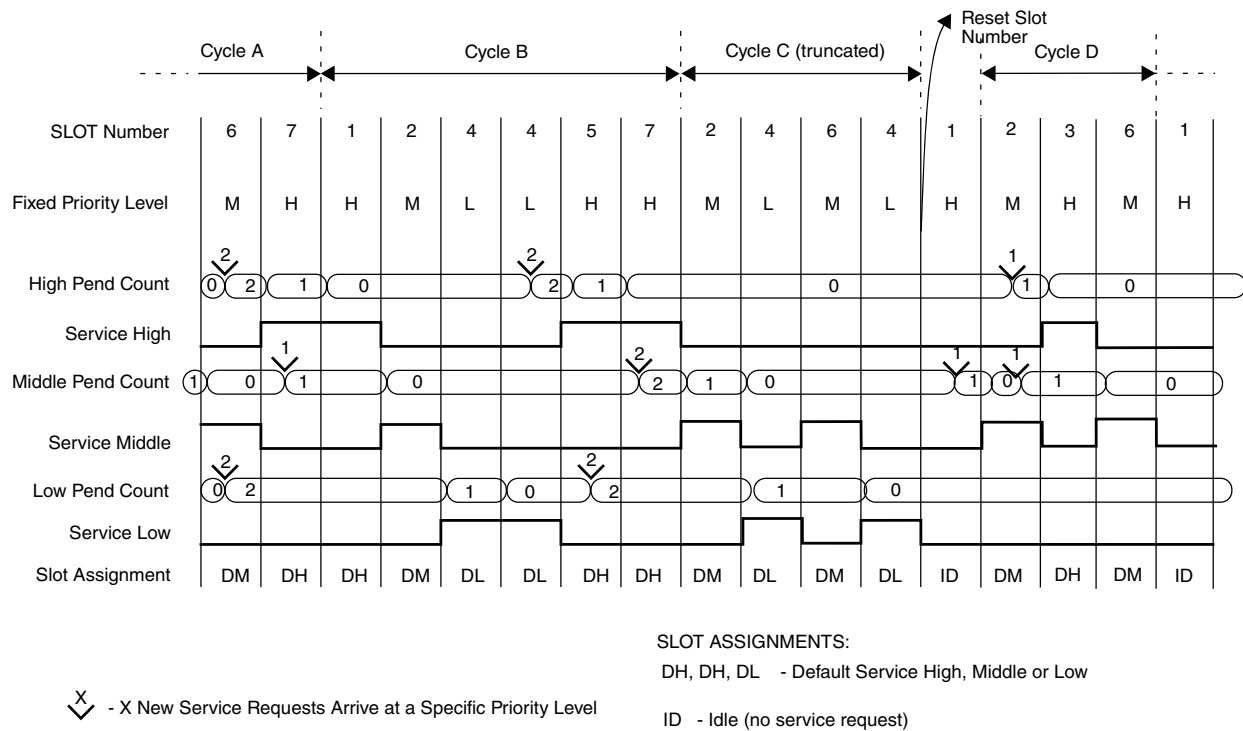


Figure 35-301. Priority Passing Disabling Example

### 35.15.3.2.3 Secondary Scheme - Priority Among Channels on the Same Level

Because channels can randomly request service, channels having the same priority level will inevitably request service simultaneously. A secondary scheme prioritizes these requests. The Scheduler services channels on each of the three priority levels, beginning with the lowest numbered channel on that level.

### 35.15.3.2.4 Priority Scheme Example

The overall priority scheme simultaneously incorporates both primary and secondary schemes. Combining both schemes in the following example conveys their correlation.

1. One high-priority and one low priority channels request service, while the Scheduler is in time slot one. Having its service request bit asserted, a single high-level channel is granted the time slot, which has high-level priority (primary scheme) and its service grant bit is asserted. At the end of the thread, the service grant bit is negated (no more requests of high priority level channels).

2. The Scheduler proceeds to time slot two, which has middle-level priority; however, no middle-level channel is requesting service. Priority is passed to the high level, but no high-level channel is requesting service; therefore, priority is passed again, and service is granted to the single requesting low-level channel. Once serviced, this channel's grant bit is negated (no more low-level requests).
3. The Scheduler resumes with the fixed-priority sequence on time slot three; however, no channels are requesting service. The Scheduler returns to time slot one, waiting for requests.
4. Two high-level and two middle-level channels simultaneously request service. Being in time slot one which is assigned high priority, the Scheduler finds the lowest numbered high-level channel (secondary scheme) and selects it for service. This channel's service grant bit is asserted.
5. The Scheduler continues to time slot two, which has middle priority (primary scheme), and allocates the slot to the lowest numbered middle-level channel requesting service (secondary scheme). The Scheduler notes the still unserviced middle-level channel and proceeds to time slot three.
6. Time slot three is allocated for high priority. The slot is allocated to the remaining unserviced high-priority channel, and the channel's service grant bit is asserted. The Scheduler checks again at the end of the thread. All service grant bits of high-level requested channels are asserted; therefore, all high-priority channels that requested have been allocated execution time. Under this condition, all service grant bits of the high-level serviced channels are negated. The Scheduler proceeds to time slot four.
7. Time slot four is allocated for low-priority channel; however, no low-level channel is requesting service. Priority is passed to the high level, but no high-level channel is requesting service; therefore, priority is passed again, and service is granted to the remaining middle-level channel which requests service. This channel's service grant bit is asserted. The Scheduler checks again at the end of the thread. All grant bits of middle-level requested channels are asserted; therefore, all middle-priority channels have been allocated execution time. Under this condition, all service grant bits of the middle-level serviced channels are negated. The Scheduler proceeds to time slot five. Meanwhile a low priority channel requests service.
8. Time slot five is allocated for high-priority channels, but there are no more requests from high-priority or middle priority channels. The single low-level channel which required service is granted time slot five. Once serviced, the channel's service grant bit is asserted. Next, the service grant bit is negated (no more requests of low priority level channels).

9. The Scheduler resumes with the fixed-priority sequence on time slot six; however, no channels are requesting service. The Scheduler returns to time slot one and waits for requests.

35.15.3.3 Time slot latency

Latency is the amount of time between a service request and the beginning of service on that channel. The following factors affect latency:

- Number of active channels
- Number of channels on a priority level
- Number of available time slots on a priority level
- Number of microcycles required to execute a thread of a Function
- Number of SDM accesses during execution of a Function thread
- eTPU clock frequency

Each time slot may require a different number of microcycles, depending on the thread of a Function to be executed. This variation is shown in the figure below.

For more details on latency evaluation, see [Estimating Worst Case Latency](#).

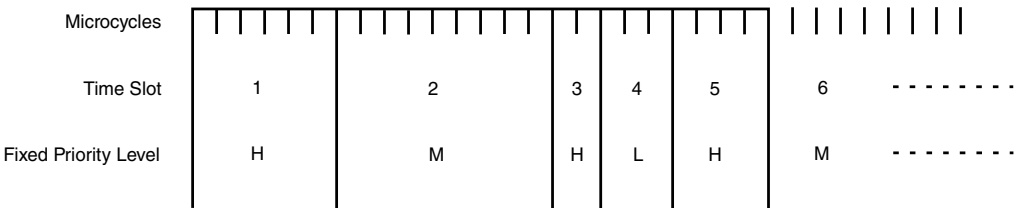


Figure 35-302. Time-slot variation

35.15.4 Parameter sharing and coherency

SDM can be concurrently accessed by Host and Microengines (two in a Dual eTPU Engine system). In general, there is no guaranteed order by which a group of parameters is accessed, which may lead to a lack of internal consistency if two or more related parameters are read when only part of them is updated.

eTPU provides mechanisms to guarantee parameter coherency. The most generic mechanisms for Host-eTPU coherency, suitable for any number of parameters, are:

- the use of Transfer Service Thread mechanism.
- the mailbox (or "software semaphore") mechanism.

These mechanisms, described in [Multiple parameter coherency methods](#), use microcode to transfer parameters from temporary buffers in SDM to their definitive locations (or vice-versa). These methods have the disadvantage of wasting processing and code memory resources.

eTPU also provides a **Coherent Dual-parameter Controller - CDC** - mechanism. It is used by Host to coherently transfer pairs of parameters from/to a parameter buffer located on SDM to/from the locations on SDM where parameters are accessed directly by the channels. Coherency is guaranteed by SDM access arbitration. Although limited to two parameters only, it has lower latency and wastes no microengine resources<sup>16</sup>. CDC usage is described in [Coherent Dual-parameter Controller \(CDC\)](#).

For parameters shared by both Engines, eTPU provides **Hardware Semaphores**. Coherency is assured given the semaphores are used to prevent concurrent access to the changing parameters. Microengine can request semaphores using specific microinstructions (see [Semaphore Operations](#)). Hardware Semaphores are described in detail in [Hardware Semaphores](#).

Neither Host nor CDC have access to the hardware semaphores, but they can be combined with microcode transfer mechanisms if Host must coherently access parameters which are also shared by both Engines.

In order to ensure coherent access to a group of parameters by two or more contenders, **each contender must have atomic access** to the shared parameters. Atomicity conditions are discussed in [Host side atomic access](#) and [Microengine Side Atomic Accesses](#).

### 35.15.4.1 Host side atomic access

Host side atomic accesses can be achieved by either of following ways:

- For one parameter, the SDM should be accessed by 32-bit-wide data transfers to ensure coherency

---

16. A microengine access to the SDM in the moment CDC is performing the transfer may suffer a maximum of two wait-states.

- For two parameters only, using the Coherent Dual Parameter Controller.
- Indirectly, for any number of parameters, by requesting microcode to coherently access SDM in its behalf. The host side atomicity problem becomes, then, a microengine side atomicity problem. Some methods that use this approach to achieve coherency are described in [Multiple parameter coherency methods](#).

## 35.15.4.2 Microengine Side Atomic Accesses

### 35.15.4.2.1 Microengine Single Parameter Atomicity

SDM should be accessed by 32-bit-wide data transfers to ensure atomicity for 32-bit parameters. This applies either to Host-Microengine coherency or Microengine-Microengine coherency in a dual eTPU Engine system.

### 35.15.4.2.2 Microengine Dual Parameter Atomicity

Microengine has the ability to access two parameters coherently in back-to-back accesses, at random addresses: once it accesses SDM, it has priority over Host for another access in the next microcycle (see [SDM arbitration](#)). Note that it applies **only to Microengine-Host** coherency. For Microengine-Microengine coherency in a dual eTPU Engine system, one must use Hardware Semaphores (see [Hardware Semaphores](#)).

Microengine dual back-to-back accesses are guaranteed to be atomic in relation to Host slave accesses or Coherent Dual-parameter Controller, regardless of semaphore usage: Host or CDC accesses cannot break-up a back-to-back Microengine access, neither Microengine can break a CDC transfer, due to the SDM arbitration mechanism described in [SDM arbitration](#).

Atomicity is not guaranteed if microengine enters halt state in the middle of a back-to-back access (see [Microengine Halt State](#)): Host can access SDM while microengine is halted in the middle of a back-to-back access.

### 35.15.4.2.3 Microengine Side Multiple Atomicity

Hardware Semaphores must be used for Microengine-Microengine coherency (more than 1 parameter) since two or more accesses from one Microengine are not atomic with respect to the other.

For multiple Microengine-Host coherency, the software methods described in [Multiple parameter coherency methods](#), or similar ones, must be used. Some of these methods rely on the fact that parameter access of a thread is atomic in relation to another thread in the same Engine, since a thread cannot be suspended (preempted).

For 1 parameter coherent access, or dual parameter coherency between only one Microengine and Host, the alternatives shown in previous sections apply.

### 35.15.4.3 Coherent Dual-parameter Controller (CDC)

Dual-parameter coherency is supported by a Coherent Dual-parameter Controller hardware (CDC), which contends with microengine for SDM access. CDC atomically transfers, upon Host's command, two parameters from one area of the SDM to another. One area is a temporary (buffer) area, where the two parameters are directly read or written by the Host. This temporary area has to begin in an SDM address multiple of 2 words, and the two parameters must be sequential. The other area is the channel parameter area where the microcode normally accesses the parameters, usually with the channel relative address mode (see [SDM Addressing Modes](#)). In this area, the parameters transferred by CDC don't have to be sequential. A transfer from the temporary area to the channel area, when the Host sends data to the channel, is called a **write transfer**. Inversely, in a **read transfer** the parameters are copied from the channel area to the temporary area (channel to Host).

Coherency is guaranteed by the SDM access contention rules implemented in the SDM arbiter (see [SDM arbitration](#)). CDC transfers are coherent in respect to the two Engines, so the target parameters in the channel area may be shared by channels on them both. During CDC operation, the Host may suffer from 3 up to 11 eTPU clocks wait states<sup>17</sup>, and the Microengine(s) may suffer up to 2 microcycle wait-states<sup>18</sup>. CDC accesses are atomic with respect to Microengine(s) accesses to the SDM. Even when neither engine is in TST, CDC may suffer up to 4 eTPU clock internal wait-states from SDM arbiter, meaning 9 slave wait-states to Host, so that it does not break atomic back-to-back accesses from microengine(s). CDC also cannot break TST preload accesses. Host can initiate CDC back-to-back transfers: there is no need of idle slave cycles between two transfers.

17. The maximum number of Host wait states on CDC occurs when both microengines overlap their TSTs, delayed 3 eTPU clocks from each other.

18. One microcycle takes two eTPU clocks. Microengines get wait-states in multiples of microcycles, while Host and CDC wait-states are multiples of eTPU clocks.



## Note

SDM memory error detection and correction does not affect the number of wait-states to host due to CDC (see [Memory error support](#)).

CDC can be used to atomically fix SDM soft errors. For more details, see [Error fixing](#).

### 35.15.4.3.1 CDC programming

The Coherent Dual-parameter Controller Register (see [eTPU Coherent Dual-Parameter Controller Register \(eTPU\\_SCR\\_CDCR\)](#)) is used to configure and initiate CDC transfers between the temporary area and channel parameter area. Host asserts STS bit in order to start the data transfer. CDC then contends for the SDM and starts the transfer. When the data transfer is complete, STS returns to 0. Host receives wait-states for writing STS=1 while CDC contends for SDM and during the transfer. The write access ends when CDC finishes the transfer. Host receives wait-states during the CDC transfer. If Host writes [eTPU Coherent Dual-Parameter Controller Register \(eTPU\\_SCR\\_CDCR\)](#) with STS=0 or does not write the STS byte, the CDC transfer does not occur. CDC programming can be summarized as follows:

1. If it is a write transfer, i.e., from Host to channel, write the two parameters into temporary area.
2. Write **STS** =1 and the remaining CDC programming parameters: parameter width (32 or 24 bits, field PWIDTH), transfer direction (read or write, field WR), temporary parameter area base address (field PBBASE), and the absolute addresses of the parameters to be transferred (concatenation of the fields CTBASE and PARM0/1).
3. If it is a read transfer -- that is, from channel to host -- read the two parameters from the temporary area into Host memory/registers.

### 35.15.4.4 Hardware Semaphores

eTPU provides Hardware Semaphores accessible by the Microengine only. It is the responsibility of the application to ensure proper use of the semaphores (i.e., agree upon a specific semaphore and use it properly, to ensure coherency).

The eTPU microinstruction set has support for locking and freeing the semaphores, described in [Semaphore Operations](#), and this is the only way to access them.

There are four semaphores available, which reduces the amount of collisions by assigning unrelated data transfers to different semaphores. Semaphores are used for parameters which can be shared by channels in different Engines, and for Engine-to-Engine synchronization. Semaphores are also the only way to ensure coherent access to parameters shared between the two Microengines.

Attempting to lock one semaphore (even not successfully) frees the other locked by the same Engine, ensuring one can lock just one semaphore at a time. That prevents deadlock conditions between the two Engines.

Microcode END command or Engine being in idle state (no thread executing) automatically releases all semaphores from one Engine side, even if a semaphore lock is done in parallel. However, it is recommended to write the microcode in a way which locks semaphores for the shortest required period, and frees them without waiting for the END command, to improve the performance of the other microengine. Semaphores are free after reset. An Engine can only free a semaphore locked by itself.

Semaphore lock requests are always **non-blocking**, in the sense that they do not suspend the requester in case the semaphore is already locked. The semaphore status after the lock request - indicating if it was successfully locked or not - must be tested through the SMLCK microengine branch condition (see [Branch conditions](#)).

### **35.15.4.5 SDM arbitration**

Up to four entities can access SDM:

- Two Microengines (in a dual eTPU Engine system)
- The Coherent Dual-parameter Controller (CDC)
- The Host CPU (direct memory-mapped access)

The following rules specify the access priorities for contended access. They keep compatibility with the TPU3 dual parameter access atomicity, but only between the microengine and CDC (not Host accesses through slave bus).

1. Microengine accesses from the two eTPU Engines are interleaved between each other, but not with Host or CDC accesses;
2. The eTPU microengine(s) gives priority for SDM accesses to either the Host CPU or the CDC under any of the following conditions:
  - a. The microengine has completed accessing the second parameter in a back-to-back SDM access<sup>19</sup>.

- b. The SDM was not accessed during the last arbitration slot for the microengine and the host does not lose the access to the other engine in the current arbitration slot<sup>20</sup>
  - c. CDC is transferring data, after its first (read) access. Note that the CDC can be in middle of a data transfer of another pair of parameters, unrelated to the ones that microengine tries to access.
3. The eTPU microengine takes priority for SDM accesses under either of the following conditions:
- a. the Host CPU or CDC has done a data transfer during the last access arbitration slot for the engine<sup>20</sup>. Also, the Host CPU does not hold a pending access against the other eTPU microengine.
  - b. the microengine is arbitrating for the access of its second parameter in a back-to-back access<sup>19</sup>. All pairs of back-to-back parameter accesses are coherent with respect to Host and CDC (not to the other microengine).

The direction (read or write) of any individual access by Host or microengine is irrelevant to the arbitration. The use of Normal or PSE SDM area by the Host is also irrelevant to the arbitration.

The first parameter preloading in a TST is considered first access by the arbiter, regardless of any access made at the END microinstruction of the previous thread, that is, the last access of a thread and the first preload are never considered a back-to-back access. On the other hand, the TST preload accesses are considered back-to-back and are, therefore, atomic with respect to Host or CDC.

### Note

The Zero SDM operation (see [Zero SDM Operation](#)) is considered an SDM access for arbitration purposes both on writes and reads; the fact that read SDM data is discarded is irrelevant for arbitration.

19. If microengine tries to access the SDM in the following microcycles, the third and fourth consecutive accesses are considered the first and second of a new back-to-back dual access.

20. The microengine access slot is between its own T4 and T2 edges (see [Microcycle and I/O timing](#)).

## 35.15.5 Enhanced Channels

Enhanced Channels comprise hardware support for input digital signal processing and output signal generation. Each Channel is associated with one input and one output signal. Enhanced Channel logic is combined with Function microcode (and optionally Angle Mode logic) to implement Channel I/O functionality.

eTPU's Enhanced Channels are capable of **dual action**, meaning that each channel logic can handle two events at different times and/or cause two separated actions - these actions and events can be mutually dependent (with the first either blocking or enabling the other), or both independent, depending on the programmed **Channel Mode**.

Each Enhanced Channel contains event logic containing two **Event Register sets**, each set supporting one input and/or output action, the pair implementing dual action support. Each Event Register set contains two 24 bit registers: Match and Capture. The **Match register** holds the pending match value which is compared against one of the two time bases by an equal-only/greater-equal comparator. The **Capture register** captures one of the two time bases as a result of a Match or Transition detection. Service Requests are issued on particular combination of match and capture events, defined by the selected Channel Mode.

In the context of the eTPU channels, a **Match** is a comparison between a time base value and a channel Match register. If those two values are coincident, or the time base value is greater than the value of the Match register, a **Match Event occurs**. Depending on the channel mode of operation and current state of the channel logic, the match event may be recognized, i.e., change the state of the channel, or be ignored. A match event recognized by the channel logic is called a **Match Recognition**. Match Recognitions can cause, also depending on Channel Mode and current state, the channel to request service, configuring a **Match Service Request**.

eTPU uses two kinds of comparator to assert a Match Event: an **Equal comparator**, in which both the Match Register and the value of the selected time base must match exactly, and a **Greater-Equal comparator**. The Greater-Equal comparator considers any time base value between the range  $[N : N+0x800000-1]$  as a valid match against the value of N in the Match Register, even when the value  $N+0x800000-1$  wraps around the point of origin (0x0). See the following figure for an illustration of the matching values on a Greater-Equal comparator.

The second source of events for the eTPU channel is a **Transition** detected at the corresponding channel's input signal. Two distinct Transition detections can be programmed individually for each channel, allowing recognition of all possible combinations of edge detection. It is also possible to check the sampled state of an input

signal upon the occurrence of a Match: the sampling of the expected value is treated as a Transition, even if the input signal did not necessarily toggled at the time of the Match, or at any time at all.

Like Match Events, **Transitions Events** may or not be recognized by the channel logic. When they are, a **Transition Detection** occurs. As well as Match Recognitions, Transition Detections can issue a Channel Service Request, depending on Channel Mode and current state.

Transition Detections and Match Recognitions are sometimes simply called Transitions and Matches throughout this document, for short.

Input and output signals can be processed separately by the channel logic and microcode, and can also be combined such that Matches and Transitions are used to cause **output signal actions**. The output signals can also be directly controlled by microcode. Many event combinations are allowed for a channel, given the possibility of configuring pairs of matches and transitions for the dual action logic, where each event is able to block or enable the next one. There is a full set of Channel Modes described in [Channel Modes](#), exploring all the capabilities mentioned here.

Each channel has its own set of registers and flags. They are selected, and made accessible to the Microengine, according to the value written into the microengine **CHAN Register** that points to the desired channel. Every time the CHAN register is written, even if with the same previous value, a channel is selected and its flags and registers are updated. For further detail, see the Channel Selection Register-CHAN topic in the General Channel Registers section.

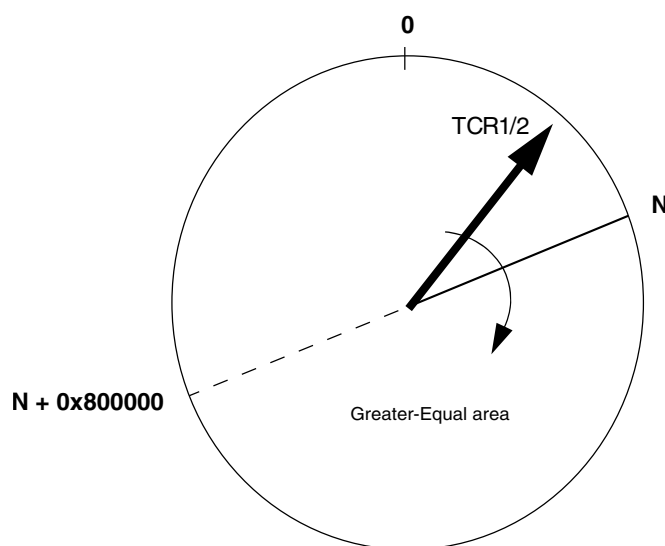


**Figure 35-303. Greater-Equal Comparator**

Beyond the request of services due to the signal and timing internal to each channel, one eTPU channel microcode can explicitly request service from another channel through the microengine **LINK Register**. A microcode write to the LINK Register asserts a service request to the channel whose number matches the contents of LINK. See [Channel Link](#) for a complete description of this mechanism.

These service requests originated in the eTPU Enhanced Channels (either time base match, input signal transition, or link service request) result in a call to the corresponding channel service routine, which is the sequence of microinstructions that is called a **Thread**. For further detail, see [Functions and threads](#).

In addition to Event Logic, each Channel has an Output Buffer Enable signal, controlled by microcode, and an Enhanced Digital Filter, which eliminates spurious glitches on input pin signal. See the chip-specific eTPU information for how the Output Buffer Enable signal is controlled.

A high level diagram of Channel logic and registers is shown in the following figure.

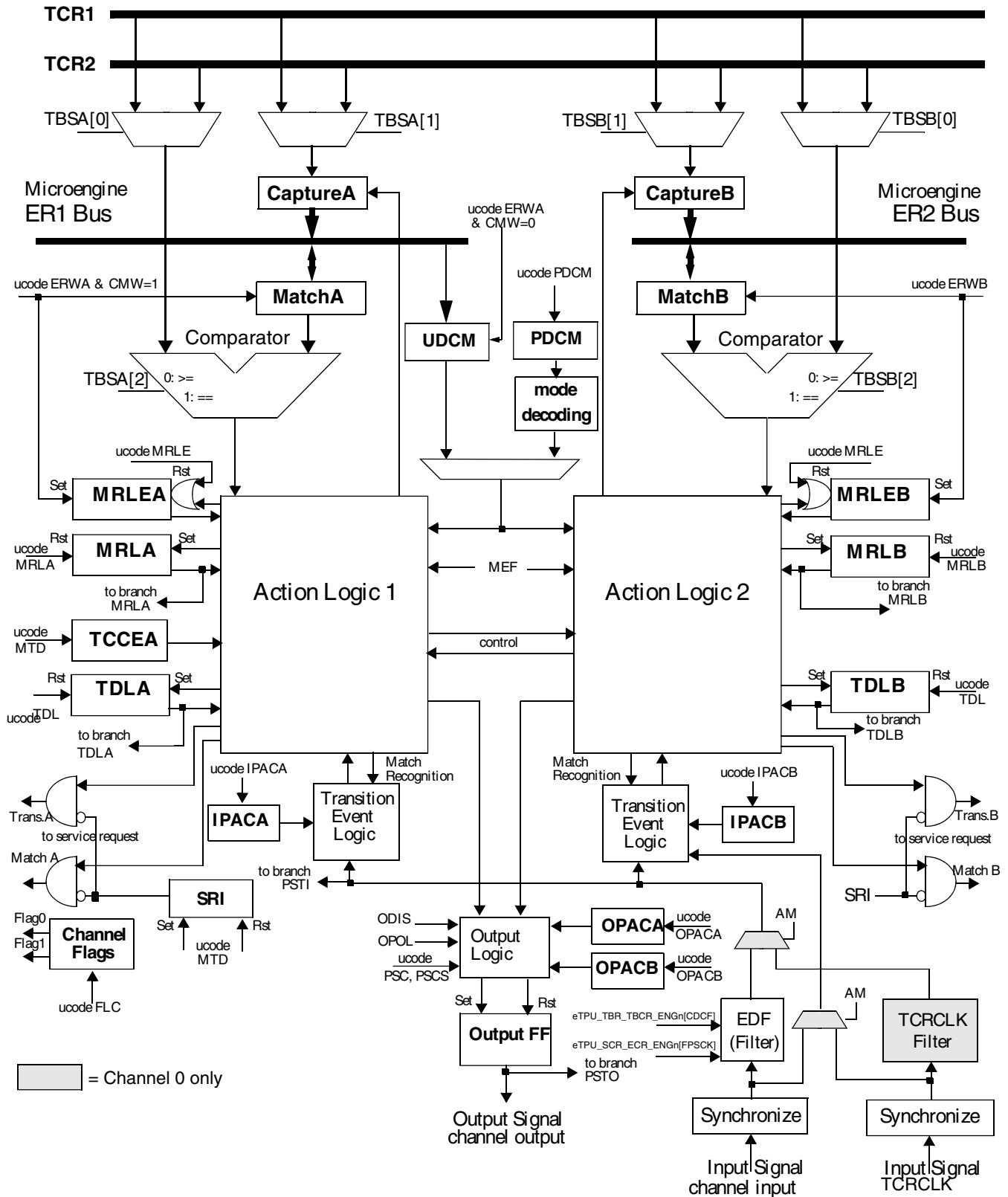


Figure 35-304. Channel logic block diagram

### 35.15.5.1 Channel registers and flags

Channel configuration and control registers can be divided in the following groups:

- **Host Configuration and Control registers**, which define channel Function and parameter allocation in SDM, input signal filtering, manage Host interrupts, and are used for Host Service Requests; they can only be accessed by Host, except for the Function Mode bits which can be also tested by microcode.
- **Event Registers**, which can only be accessed by eTPU Microengine, through dedicated Channel Control microinstruction operations (see [Channel Control and Configuration Microoperations](#)); these registers are directly used to implement channel functionality, and include channel event status latches which can be directly tested by Microengine branch instructions.
- **Pin Control registers**, which basically define pin state and transition polarity (but not input signal filtering); they are accessible only by dedicated Channel Control microinstruction operations.
- **Link registers**, which implement the channel link mechanism that allows one channel to request service to another one; they are accessible only by microinstruction operations.
- **General Channel registers**: CHAN, SRI, Flag0/1, PDCM, UDCM.

Most of those registers are channel exclusive, i.e., there is one copy of them for each channel. Microcode can access registers from only one channel at a time. The Channel Selection (CHAN) register (see [Channel Selection Register - CHAN](#)), accessible only by microcode, defines the channel whose registers are being accessed, with exception of link register and function mode. CHAN register assumes the value of the channel to be serviced at the beginning of TST.

The Service Request Inhibit - SRI - register controls the generation of Service Requests on matches and transitions, also affecting channel logic behavior. For a full description see [SRI - Match/Transition Service Request Inhibit Latch](#). Flag0/1 are used to select channel service threads based on channel software state. See [Flag1,Flag0 - Channel "state resolution" flags](#) for more details.

Host Configuration and Control registers are described in [Channel Configuration Control memory map and registers](#).

Time Base configuration is common to all channels, and described in [Time bases](#). Time Base selection for matches and captures, however, is individual to each channel, and is part of the Event Registers.

Link registers are described in [Channel Link](#).



The following sections describe the Event Registers and Pin Control registers.

### 35.15.5.1.1 ER - Event Registers

Each channel contains two identical Event Register sets, named ERA and ERB, corresponding to the two actions supported. Each Event Register set contains:

- a 24-bit Match register (Match A or Match B), which holds a match value. This value is compared against the selected match time base (TCR1 or TCR2).
- a 24-bit Capture register (CaptureA or CaptureB), which samples the selected capture time base (TCR1 or TCR2)
- a Time Base Selection register (TBSA or TBSB)
- a Match Recognition status flag (or latch) (MRLA or MRLB)
- a Match Recognition Enable latch (MRLEA or MRLEB)
- a Transition Detection flag (or latch) (TDLA or TDLB)
- a Transition Continuous Capture enable (TCCEA only)

ERA and ERB are associated with the first and second events in double action modes, not necessarily in that order. The order of Match events associated with ERA and ERB depends on the programmed channel mode, the MatchA and MatchB values, and the timebases selected by TBSA and TBSB. Similarly, the order of Transition events associated with ERA and ERB depends on the programmed channel mode, and the transition detection selected by IPACA and IPACB.

These registers are directly or indirectly accessed by the microcode. TBSA and TBSB registers are defined in [TBSA and TBSB - Time Base Selection Registers](#). The other registers are explained in [Match Recognition](#) and [Transition detection and Time Base Capture](#).

Access to the Event Registers is qualified by the channel currently selected by the microengine (i.e., the channel value currently in the CHAN register). During the channel transition period (automatic CHAN assignment), or whenever CHAN is written by microcode, Capture values of the new selected channel are sampled into Microengine registers ERTA and ERTB, therefore becoming visible to the microcode. At the same time, updated values of MRLA, MRLB, TDLA and TDLB are sampled into the branch logic, making the register values and the flags coherent with respect to each other and with the thread selected by the Scheduler. <sup>21</sup>

---

21. The thread selected is determined by the Entry Point which, in turn, is determined partially by the channel latches. See [Entry Point address generation](#).

## Note

The Function Mode bits are also sampled from the Host interface on Time Slot Transition, so that they remain constant to microengine even when Host changes them.

During service, the microcode can access updated values of the Event Registers of any channel by writing the channel number to CHAN. Writing CHAN with the same value (CHAN := CHAN) updates ERTA and ERTB with the new captured values, the branch logic with updated MRLA/B and TDLA/B flags. Writing CHAN with a different value does the same with the values from the newly selected channel.

Match values are also accessed through ERTA and ERTB Microengine registers, which are copied to/from the channel MatchA and MatchB registers by specific microinstruction operations.

Microcode writes to the flags and selections (MRLA/B, TDLA/B and TBSA/B) are immediately effective to the channel. The MRLA/B and TDLA/B branch conditions are also immediately reset when their correspondent flags are reset by microcode. Match registers are indirectly written by microcode through ERTA/B. MRLEA/B is unconditionally asserted when respective Match register is updated from ERTA/B, and its negation is immediate.

The following table summarizes Event Registers accesses.

**Table 35-338. Event Registers microcode accesses**

| Register           | Access type                                                                           | Sampled from channel         | Update to channel        | Microcode fields <sup>1</sup>    | Reset value <sup>2</sup> |
|--------------------|---------------------------------------------------------------------------------------|------------------------------|--------------------------|----------------------------------|--------------------------|
| CaptureA, CaptureB | read through ERTA/B                                                                   | to ERTA/B on CHAN assignment | no                       | T2ABD                            | n.a.                     |
| MatchA, MatchB     | read and write through ERTA/B                                                         | to ERTA/B by microcode       | from ERTA/B by microcode | ERWA, CMW, ERWB, T4ABS           | n.a.                     |
| MRLEA, MRLEB       | write to 0 (negate) directly;<br>write to 1 (assert) upon MatchA/B update from ERTA/B | no                           | immediate                | MRLE, ERWA, ERWB                 | 0, 0                     |
| TBSA, TBSB         | write only                                                                            | no                           | immediate                | TBSA, TBSB                       | 000, 000                 |
| MRLA, MRLB         | flag test on branch,<br>write to 0 (negate) only                                      | on CHAN assignment           | immediate                | BCC (test)<br>MRLA, MRLB (reset) | 0, 0                     |
| TDLA, TDLB         | flag test on branch,<br>write to 0 (negate) only                                      | on CHAN assignment           | immediate                | BCC (test)<br>TDL (reset)        | 0, 0                     |
| TCCEA              | write only                                                                            | no                           | immediate                | MTD                              | 0                        |

1. See [Microinstruction Set](#).
2. n.a. means that value of the register is undetermined after reset.

### 35.15.5.1.1.1 MatchA and MatchB Registers

MatchA and MatchB registers hold a match value to be compared against the selected channel time base. A match value can only be written into the Match register by microcode, through ERTA/B microengine registers (see [Write Channel Match and UDCM Registers](#)). Microcode can also read the Match register as a special T4ABS source operation, when T4ABS=0101, and the source for T4ABS is selected from the second register set. In this operation, MatchA/B registers are copied into ERTA/B registers (see [Selecting sources and destination](#)). For more information on time base matches, see [Match Recognition](#).

### 35.15.5.1.1.2 CaptureA and CaptureB Registers

CaptureA and CaptureB registers capture the selected channel time base. CaptureA/B registers cannot be directly written or read by microcode. During the Time Slot Transition (TST) or during CHAN assignment, CaptureA/B registers are copied into ERTA/B microengine registers. For more information, see [Transition detection and Time Base Capture](#).

### 35.15.5.1.1.3 TBSA and TBSB - Time Base Selection Registers

TBSA/B are 3-bit registers which have the following effect on channel configuration:

- Selection of the timebase (TCR1 or TCR2) to be compared against the match values in MatchA and/or MatchB registers.
- Selection of the timebase (TCR1 or TCR2) to be captured in the CaptureA and/or CaptureB registers by a match or transition detection event.
- Selection of comparator mode to be used with MatchA and MatchB registers: equal-only or greater-equal.

After reset TBSA/B are 000. The following table shows values of TBSA and TBSB for configuration selection. Note that the time base selection for capture is independent of the time base selected for matches.

**Table 35-339. TBSA/B Programming States**

| TBS bit   | 2                    | 1                           | 0                         |
|-----------|----------------------|-----------------------------|---------------------------|
| bit value | Comparator selection | Capture time base selection | Match time base selection |
| 0         | greater or equal     | TCR1                        | TCR1                      |
| 1         | equal-only           | TCR2                        | TCR2                      |

TBSA/B are written through the microcode fields TBSA/B (see [Comparator and Time Base Selection](#)).

#### 35.15.5.1.1.4 MRLA/B - Match Recognition Latches

See [MRLA/B - Match Recognition Latches](#).

#### 35.15.5.1.1.5 MRLEA/B - Match Recognition Latch Enable

See [MRLEA/B - Match Recognition Latch Enable](#).

#### 35.15.5.1.1.6 TDLA/B - Transition Detection Latch

See [TDLA/B - Transition Detect Latches](#).

#### 35.15.5.1.1.7 TCCEA - Transition Continuous Capture Enable

See [TCCEA - Transition Continuous Capture Enable](#).

### 35.15.5.1.2 Pin Control Registers

Pin Control Registers are replicated one per channel, accessed only by microcode and qualified by the CHAN register in the same way as Event Registers.

The following table lists Pin Control Registers, explained in following subsections, and their accesses.

**Table 35-340. Pin Control Registers microcode accesses**

| Register     | Access type                | Sampled from channel | Update to channel | Microcode fields <sup>1</sup> | Reset value |
|--------------|----------------------------|----------------------|-------------------|-------------------------------|-------------|
| IPACA, IPACB | write only                 | no                   | immediate         | IPACA, IPACB                  | 000,000     |
| OPACA, OPACB | write only                 | no                   | immediate         | OPACA, OPACB                  | 000,000     |
| PSTI         | flag test on branch        | no                   | no                | BCC                           | 0           |
| PSTO         | flag test on branch, write | no                   | immediate         | BCC (test)                    | 0           |

*Table continues on the next page...*

**Table 35-340. Pin Control Registers microcode accesses (continued)**

| Register         | Access type         | Sampled from channel | Update to channel | Microcode fields <sup>1</sup> | Reset value       |
|------------------|---------------------|----------------------|-------------------|-------------------------------|-------------------|
|                  |                     |                      |                   | PSC, PSCS (set)               |                   |
| OBE <sup>2</sup> | write only          | no                   | immediate         | TBSA values 1000,1001         | 0<br>(negated)    |
| PSS              | flag test on branch | on CHAN assignment   | no                | BCC                           | n.a. <sup>3</sup> |
| PRSS             | flag test on branch | on CHAN assignment   | no                | BCC                           | n.a. <sup>4</sup> |

1. See [Microinstruction Set](#).

2. Although this register exists for code compatibility reasons, OBE functionality is not available on this chip.

3. PSS is PSTI or PSTO sampled on CHAN assignments and at thread start.

4. PRSS is PSTI sampled on channel service request.

### 35.15.5.1.2.1 IPACA/B and OPACA/B - Input and Output Pin Action Control Registers

These registers determine the transition detections and output pin actions which takes place due to match or transition events. Each field is three bit wide. After reset, the IPACA/B and OPACA/B registers are set to 000. IPACA and IPACB registers are mutually independent and have identical encoding, and so do OPACA and OPACB. The table below shows IPAC and OPAC encoding. Note that transition detections are independent from the output actions, but some options of output actions are triggered by transition detections. Output actions can also be triggered by Matches. For a detailed definition of Transition Detections, see [Transition detection and Time Base Capture](#).

IPACA/B define the Transition events treated by channel logic. Although the name "transition" is generically used for the transition detections, IPAC options 100 and 101 do not really detect transitions: they actually sample the state of the input signal at the occurrence of the corresponding Match (Match A used for IPACA, Match B used for IPACB).

**Table 35-341. IPACA/B and OPACA/B Encoding**

| Value | IPAC meaning                                  | OPAC meaning                                 |
|-------|-----------------------------------------------|----------------------------------------------|
| 000   | Do not detect transitions                     | Do not change output signal                  |
| 001   | Detect rising edge only                       | Match <sup>1</sup> sets output signal high   |
| 010   | Detect falling edge only                      | Match <sup>1</sup> sets output signal low    |
| 011   | Detect either edge                            | Match <sup>1</sup> toggles output signal     |
| 100   | Detect input signal = 0 on Match <sup>1</sup> | Transition sets output signal low            |
| 101   | Detect input signal = 1 on Match <sup>1</sup> | Transition detection sets output signal high |
| 110   | reserved                                      | Transition detection toggles output signal   |
| 111   | n.a. <sup>2</sup>                             | n.a. <sup>2</sup>                            |

1. Match A is used for IPACA/OPACA, and Match B for IPACB/OPACB.
2. On the microinstruction fields IPACA/B and OPACA/B this value is neutral, meaning that IPAC/OPAC register values are not changed.

### 35.15.5.1.2.2 Output Pin Control Logic and Pin State Output Register - PSTO

The output signal control logic uses OPACA/B, the Pre-defined Channel Mode (PDCM) and the User Defined Channel Mode (UDCM), and the microcode Pin State Control (PSC and PSCS) fields. It is responsible for setting the Pin State Output (PSTO) register to the specified logic value required by microcode, by input events, or by Match A and/or Match B events. The PSTO register stores the driven pin state determined by the Pin Control logic. The Output Buffer Enable signal, if used at MCU integration, must be set by microcode (using TBSA field) to make the pad propagate the PSTO register output to the actual pin.

PSTO register output also goes to the microengine branch logic, enabling branching on the driven pin state. PSTO is set to 0 on reset.

The PSC and PSCS microcode fields are used for setting the PSTO register to a fixed value, or to the value specified by the OPACA or OPACB microcode field, as shown in the following table.

**Table 35-342. PSC and PSCS encoding**

| PSC | Output pin action                                              |
|-----|----------------------------------------------------------------|
| 00  | Force pin state according to OPACA (PSCS=0) or OPACB (PSCS=1). |
| 01  | Force pin high.                                                |
| 10  | Force pin low.                                                 |
| 11  | Do not change pin state.                                       |

See [Channel Control and Configuration Microoperations](#) for details.

On occurrence of match recognitions or transition detections, the pin control logic sets PSTO value according to the event number (Match A/Transition A or Match B/Transition B) and the contents of OPACA/IPACA or OPACB/IPACB registers. There are cases in which two match or transition events may occur at the same time, each of them trying to force a different pin action. The resolution of the selected match event which sets the value depends on the Pre-defined Channel Mode (PDCM) register and the User Defined Channel Mode (UDCM). For details refer to [Match/Transition Pin Action Conflict Resolution](#).

### 35.15.5.1.2.3 PSTI and PSS - Pin State Input and Pin Sampled State Registers

During the time slot transition period, or whenever the CHAN register is written by microcode, the filtered<sup>22</sup> input signal PSTI or output signal PSTO (selected by the ETPD field in [eTPU Channel n Configuration Register for Engine 1](#) (eTPU\_CCCR\_CnCR\_ENG1) and [eTPU Channel n Configuration Register for Engine 2](#) (eTPU\_CCCR\_CnCR\_ENG2)) is sampled into the branch logic as the PSS flag (see [Table 35-411](#)). The microcode can then branch on either the currently driven (PSTO) or input (PSTI) pin state, or on sampled pin state (PSS, which is stable as long as CHAN does not change).

#### Note

If a transition occurs simultaneously (after the filter) with the CHAN assignment, PSS samples the new pin value. Therefore, if TDLA/B is cleared simultaneously with the assignment CHAN := CHAN, the occurrence of a transition at this very moment can still be tested with PSS.

### 35.15.5.1.2.4 PRSS - Pin Request Service Sample

Channel logic can, depending on its state and programmed mode, request service to the eTPU microengine (see [Functions and threads](#)). When the channel logic issues a service request, the filtered input signal PSTI is sampled into an internal channel flag PRSS. There is one such PRSS flag for each channel (see [Pin Control Registers](#)). Channel PRSS keeps its value until all its service request sources are cleared and until a new channel service request rises.

#### NOTE

Only channel service requests update PRSS. It is not updated by host service requests (HSRs) or by link service requests. Moreover, a channel service request updates PRSS even if the channel is disabled (its CPR set to 0, see [eTPU Channel n Configuration Register for Engine 1](#) (eTPU\_CCCR\_CnCR\_ENG1) and [eTPU Channel n Configuration Register for Engine 2](#) (eTPU\_CCCR\_CnCR\_ENG2)).

The channel PRSS flag is sampled into the branch logic as the PRSS flag (see [Table 35-411](#)) during the time slot transition period, or whenever the CHAN register is written by microcode.

22. The filter can be bypassed.

35.15.5.1.2.5 OBE - Output Buffer Enable control latch

NOTE

Output Buffer Enable is not available on this chip.

OBE latch drives the Output Buffer Enable signal, which can be used to control the output signal pad driver. Channel output comes disabled from reset. If an Output Buffer Enable signal<sup>23</sup> from eTPU is used on MCU integration and a signal is desired to be output in a channel, OBE signal must be set by microcode. Microcode field TBSA is used to set/reset the Output Buffer Enable control when microcode field TBSA bit 3 is 1, according to the following table.

Table 35-343. TBSA Output Buffer control

| microcode<br>TBSA[2:0] | meaning when TBSA[3]=1 |
|------------------------|------------------------|
| 000                    | enable output buffer   |
| 001                    | disable output buffer  |
| 111                    | do nothing             |
| other values           | reserved               |

23. There is one independent OBE signal for each channel.





### 35.15.5.1.3 General Channel registers

These registers control other aspects of channel logic. Except for CHAN, they are unique per channel. The following table summarizes the registers and access options.

### Table 35-344. General Channel registers microcode access

| Register | Access type | Sampled from channel | Update to channel | Microcode fields <sup>1</sup> | Reset value                                  |
|----------|-------------|----------------------|-------------------|-------------------------------|----------------------------------------------|
| CHAN     | read/write  | n.a. <sup>2</sup>    | n.a. <sup>2</sup> | T4ABS, T4BBS, T2ABD           | defaults to serviced channel at thread start |
| PDCM     | write only  | no                   | immediate         | PDCM                          | 1100                                         |

*Table continues on the next page...*

**Table 35-344. General Channel registers microcode access (continued)**

| Register     | Access type             | Sampled from channel | Update to channel      | Microcode fields <sup>1</sup> | Reset value                            |
|--------------|-------------------------|----------------------|------------------------|-------------------------------|----------------------------------------|
|              |                         |                      |                        |                               | (sm_st)                                |
| UDCM         | write only              | no                   | from ERTA by microcode | CMW, ERWA                     | parameter value defined at integration |
| SRI          | write only              | no                   | immediate              | MTD                           | 1                                      |
| Flag1, Flag0 | branch flag test, write | no                   | immediate              | BCC, FLC                      | 0, 0                                   |

1. See [Microinstruction Set](#).

2. CHAN is common to all channels in the Engine.

### 35.15.5.1.3.1 Channel Selection Register - CHAN

CHAN is the register that holds the number of the channel that qualifies the context of most Channel Registers, including Pin Control and ER accesses, and is common to all channels in a same Engine.

When a thread starts to be executed, the contents of CHAN register is automatically updated on Time Slot transition to the number of the channel to be serviced. Serviced channel is constant during channel servicing, but the selected channel can be changed any time by microengine writing into CHAN register.

Some microinstructions are affected by the serviced channel instead of CHAN. These are:

- Conditional branch using LSR (see [LINK Register](#)) or Function Mode (see [eTPU Channel n Status Control Register for Engine 1 \(eTPU\\_CCCR\\_CnSCR\\_ENG1\)](#) and [eTPU Channel n Status Control Register for Engine 2 \(eTPU\\_CCCR\\_CnSCR\\_ENG2\)](#)).
- Negate channel flag LSR, Interrupt CPU and Data Transfer Request (see [Channel Interrupt and Data Transfer Requests](#)).

When CHAN register is written, accesses are qualified by the new CHAN register value from the instruction following CHAN assignment on, except CaptureA/B sampling into ERTA/B and Match register writing from ERTA/B (see [CHAN assignment, Read Match and ERWA/B](#)).

Writing CHAN (including with the same value, CHAN:= CHAN) updates ERTA and ERTB with the new captured values, the branch logic with updated MRLA/B and TDLA/B flags.

The following table shows the commands, flags and registers selected by the CHAN register value.

**Table 35-345. CHAN-selected features**

| Feature used                                                                                               | Selected by CHAN |
|------------------------------------------------------------------------------------------------------------|------------------|
| Channel-relative SDM access                                                                                | YES              |
| Branch using PSS, PRSS, PSTI and PSTO channel flags                                                        | YES              |
| Branch using MRLA/B, TDLA/B, Flag0/1 <sup>1</sup>                                                          | YES              |
| Branch on all other conditions <sup>2</sup>                                                                | no               |
| ERTA/B Value                                                                                               | YES              |
| Configure (selected) channel                                                                               | YES              |
| Channel commands applied to: MRLA/B, TDLA/B, TBSA/B, IPACA/B, OPACA/B, PSC, PSCS, OBE, PDCM, MRLE, Flag0/1 | YES              |
| Channel command: set/reset SRI                                                                             | YES              |
| Channel command: write to Match registers (ERWA/B)                                                         | YES <sup>3</sup> |
| Channel command: read Match registers into ERTA/B                                                          | YES              |
| Clear LSR                                                                                                  | no               |
| Channel interrupts and data transfer requests (CIRC)                                                       | optionally       |

1. In TPU, these conditions retained the old values.
2. See [Branch Operations](#)
3. If write Match (ERWA/B) occurs at the same time of a CHAN assignment, the channel which is target of the write is the one selected by the new CHAN value. See [CHAN assignment](#), [Read Match](#) and [ERWA/B](#).

### 35.15.5.1.3.2 PDCM - Predefined Channel Mode

PDCM determines the channel mode assigned to the channel. Channel mode defines much of the channel logic behavior, specially how matches blocks and enables transitions and vice-versa, as well as occurrence of time base captures and service requests based on matches and transitions. For a complete description see [Channel Modes](#).

PDCM is a 4-bit register set by the microcode field of the same name (see [Predefined Channel Modes](#)), and cannot be read or tested in branch instructions. The table below relates the PDCM value with channel modes. The second column specifies the mnemonic used to reference the mode, introduced in [Channel Modes Overview](#). There is one PDCM for each channel, initialized with 1100 on reset.

PDCM is also used to select the User Programmable Channel Mode. If this selection is made, the channel behavior is defined by the settings of the UDCM register (see [UDCM - User Defined Channel Mode](#)).

**Table 35-346. PDCM encoding**

| PDCM | Channel mode |
|------|--------------|
| 0000 | em_b_st      |
| 0001 | em_b_dt      |
| 0010 | em_nb_st     |

*Table continues on the next page...*

**Table 35-346. PDCM encoding (continued)**

| PDCM | Channel mode              |
|------|---------------------------|
| 0011 | em_nb_dt                  |
| 0100 | m2_st                     |
| 0101 | m2_dt                     |
| 0110 | bm_st                     |
| 0111 | bm_dt                     |
| 1000 | m2_o_st                   |
| 1001 | m2_o_dt                   |
| 1010 | User Defined Channel Mode |
| 1011 | reserved                  |
| 1100 | sm_st <sup>1</sup>        |
| 1101 | sm_dt                     |
| 1110 | sm_st_e                   |
| 1111 | n.a. <sup>2</sup>         |

1. This is the reset value, also compatible with TPU channel behavior.

2. This value is used as a neutral (do not change) value in the PDCM microinstruction field.

### 35.15.5.1.3.3 UDCM - User Defined Channel Mode

UDCM holds the control signals that define the channel logic behavior in terms of Match and Transition blocking and enabling, captures and service requests triggered by events. The register layout is defined in the figure below. The individual fields are defined in [Channel Mode Logic and Event Flags](#). There is one UDCM register for each channel, which can be independently programmed.

UDCM can only be written into the Match register by microcode, through ERTA/B microengine registers (see [Write Channel Match and UDCM Registers](#)).

|     |      |      |       |       |       |      |       |       |      |       |     |      |   |
|-----|------|------|-------|-------|-------|------|-------|-------|------|-------|-----|------|---|
| 13  | 12   | 11   | 10    | 9     | 8     | 7    | 6     | 5     | 4    | 3     | 2   | 1    | 0 |
| MSR | MCAP | M1ET | M1EM2 | M1BM2 | M2BM1 | M2BT | T1BM1 | T2BM1 | TBM2 | T1ET2 | TSR | TCAP |   |

**Figure 35-306. UDCM Register**

### 35.15.5.1.3.4 SRI - Match/Transition Service Request Inhibit Latch

SRI blocks channel service requests due to the assertion of MRLA/B and/or TDLA/B. SRI does not affect recognition of Link Service Requests or Host Service Requests, neither MRLA/B or TDLA/B microcode branch tests nor Entry Table selection<sup>24</sup>. SRI is asserted during reset and is controlled by microcode field MTD.

24. In TPU, SRI also blocked TDL and MDL branches and enabled any transition to capture time base.

To unburden the microengine, SRI asserted configures a channel "dumb" regarding the servicing of match and capture channel service requests. Even with SRI=1, TDLA/B and MRLA/B can still be asserted, and the level specified by the OPAC (Output Pin Action Control) registers will be output to the pin.

### 35.15.5.1.3.5 Flag1,Flag0 - Channel "state resolution" flags

Each channel has a pair of flags, simply called Flag0 and Flag1, that can be set/reset by microcode through microinstruction field FLC. FLC sets/resets Flag0/1 of the channel selected by CHAN. These flags can be tested by microcode, and are also used to resolve the microcode entry point for the channel service (see [Entry Points](#)). Flag0 and Flag1 are, so, typically used for fast state resolution. FLC microinstruction field also allows Flag1,Flag0 to be copied from selected bits of P register high byte, which is also meant to be used to hold application state. Flag0 and Flag1 are both zero out of reset.

## 35.15.5.2 Match Recognition

The match operation is performed every microcycle by comparing the channel MatchA and MatchB registers against the value of the TCR bus specified for each match. TCR1 or TCR2 bus is selected according to TBSA and TBSB fields. Both results have effect on the next clock cycle (see [Microcycle and I/O timing](#)).

A Match A/B event is qualified by a set of match enabling conditions to the Match Recognition Registers MRLA/B. To recognize the match, the following match enabling conditions are required:

- For IPACA/B=0xx, Match Enable Flag (MEF), qualified by the channel currently being serviced must be asserted. For IPACA/B =1xx, Match A/B is always enabled (even during Time Slot Transition - TST), regardless the state of the Match Enable Flag (MEF). See [MEF - Match Enable Flag](#) for the conditions of MEF assertion.
- Match Recognition Latch Enable 1/2 (MRLEA/B) is asserted. A match event recognition may only occur if its corresponding MRLEA/B bit is set, which only happens upon a write to a channel match register by the microcode, copied from ERTA/B. MRLEA/B is negated when the respective match occurs or, in some double match channel modes, when a match for the other Match register occurs. It ensures that the greater-equal comparison will not cause additional matches<sup>25</sup>.
- In selected modes (see [Channel Modes](#)), the particular conditions of MRL and TDL flags of the other event, i.e:

---

25. Microcode can also negate MRLEA/B.

- MRLA, TDLA enable or block MRLB;
- MRLB, TDLB enable or block MRLA.
- The respective MRL is negated.
- In selected modes (see [Channel Modes](#)), the state of its respective TDL flag.

If the Match A and/or Match B conditions are met, the channel immediately forces the pin state if specified by the appropriate OPACA/B registers (Output Pin Action Control 1/2) and, in some cases, by IPACA/B registers. Refer to [IPACA/B and OPACA/B - Input and Output Pin Action Control Registers](#).

If both Match A and Match B events occur at the same time, with conflicting pin actions, the priority over the pin action is mode dependent. For further details on pin action resolution refer to [Match/Transition Pin Action Conflict Resolution](#).

#### 35.15.5.2.1 MRLA/B - Match Recognition Latches

MRLA/B indicate the recognition of a match event detected by the comparator. They can be asserted either on T2 or T4 (see [Microcycle and I/O timing](#)). Assertion of MRLA/B issues a match Service Request in specific channel modes, depending on previous events and state of SRI. After reset MRLA and MRLB are both negated.

When MRLA or MRLB is asserted, it may change the output signal level according to the Input and Output Pin Action Control registers (refer to [IPACA/B and OPACA/B - Input and Output Pin Action Control Registers](#)). Assertion of MRLA/B causes a capture of one or two time bases, according to the selected mode capturing scheme (see [Transition detection and Time Base Capture](#))

A match recognition is self-blocking, regardless of Channel Mode: once MRLA (MRLB) has been asserted, it negates its associated MRLEA (MRLEB) register, preventing future match recognitions, until the associated match register is rewritten by microcode. The microcode has to enable new matches by updating the new match value in the MatchA (MatchB) register<sup>26</sup>. In addition, assertion of MRLA/B can block its twin MRLB/A, depending on the channel mode. In some double match blocking channel modes, Match A/B event blocks the occurrence of Match B/A in a "first win" scheme.

It is the transition from 0 to 1 in MRL that causes the Match actions, except the service request: apart from MRLEA/B negation(s) and service request, no action due to a Match occurs if MRL was already set to 1, even if the other MRL assert conditions are satisfied. However, if a Match and a microoperation negating its corresponding MRL occur at the

26. Before that, microcode should also negate MRLA (MRLB), otherwise an old match may be recognized by the scheduler and serviced as a new one

same time, MRL negation by microcode overrides its assertion, but any dependable captures and pin action occurs anyway (if MRL was already negated before), and also the negation of MRLE(s) (the respective one and, in some channel modes, the other, regardless of MRL state before). Note that MRLE must have been set before (by writing a new Match value).

### 35.15.5.2.2 MEF - Match Enable Flag

MEF is a one-bit latch that is unique for all channels in an Engine.

MEF can selectively enable assertion of MRLA/B, depending on the IPACA/B field. For IPACA/B=0xx, MEF=1 enables assertion of MRLA/B for the scheduled channel during service. For IPACA/B=1xx, Match A/B is always enabled, regardless the state of the MEF, but it still depends on the other Match recognition conditions. Matches of channels not being serviced are never disabled by MEF.

MEF is not accessible by Microengine or Host. MEF is negated for two microcycles in the beginning of the time slot transition period. After two microcycles (plus wait-states) into TST, the ME bit in the entry point is copied to MEF to allow selective enabling of MRL for each thread (refer to [Time Slot Transition](#)). MEF is asserted unconditionally soon after a thread ends.

If a channel service needs to postpone a programmed match, MEF assures that microcode wins the race against match event after time slot transition (only for IPAC=0xx).

Note that a match event may be lost during the periods when MEF is negated only if:

- the match comparator is configured for "equal-only" behavior, and
- IPACA/B=0xx, and
- TCR increments at the rate of eTPU clock divided by 2 or faster.

When the comparator is configured as "greater-equal", the match event that occurred when MEF was negated may be recognized after MEF is asserted again, due to the "greater than" condition.

### 35.15.5.2.3 MRLEA/B - Match Recognition Latch Enable

MRLEA/B is negated upon the assertion of its respective MRLA/B. In blocking match channel modes it may also be negated together with the assertion of the twin MRLB/A. The MRLEA/B bits ensure that data captured due to the first match event will not be overwritten when MRLA/B is negated: due to greater-equal comparison, the match condition continues to be true, but should not cause another capture event.



In addition to negation by local match event, the microcode can negate both MRLEA and MRLEB, to block pending matches, and also MRLA/B, individually. This action will prevent future match events from the selected channel.

Writing the MatchA/B registers by microcode to schedule the next match values sets MRLEA and/or MRLEB and enables new matches. This setting overrides the MRLE negation conditions due to channel logic or microcode (see [Channel Modes](#)). By combining write to Match A/B with MRLEA/B negation microinstructions, the microcode can negate one MRLE while asserting the other.

### Note

If the MRLE negation conditions continue after writing MatchA/B registers, the respective MRLE does not keep asserted. For instance, if MRL=1 and a match is programmed for a time value in the past during a thread with ME=1, then MRLE will be cleared soon after MatchA/B is written, even though a match does not occur (because MRL was already asserted, neither captures nor pin toggles occur).

When the match register is updated (with MRLE already asserted before) and field MRLA/B = 1 (no clear, see [Clear Transition/Match Event Registers](#)) and MRLA/B flag is zero, the eTPU behaves exactly as the TPU, that is: a match that comes concurrently with the rewrite of the match register, matching the old value, sets the MRL, as if the setting of the MRLE due to match register write had precedence over its clear by the match at that moment. After this simultaneous operation, the MRLE value stays at 1, and the captured time base value, if any, reflects the match value.

When the match register is updated (with MRLE already asserted before) and field MRLA/B = 0 (clear MRL, see [Clear Transition/Match Event Registers](#)) and MRLA/B flag is zero, the match captures will occur, the MRLA/B flag will keep negated, and MRLE will stay asserted. Special care must be taken when a match is reprogrammed on a channel running at T2/T4 timing, please see the details in the “T2/T4 Channel Timing” section.

### 35.15.5.3 Transition detection and Time Base Capture

Time Base Capture(s) occur when the value of a specified TCR is sampled into the CaptureA and/or CaptureB register. TBSA[1] and TBSB[1] select which TCR will be captured in CaptureA and CaptureB, respectively.

A capture event may occur due to either of the following events:



- The assertion condition of Match Recognition Latch (MRL), even if MRL is simultaneously negated by microcode
- The assertion condition of Transition Detection Latch (TDL), even if TDL is simultaneously negated by microcode.
- Any Transition Event specified by IPACA if both the Transition Detection Latch TDLA and Transition Continuous Capture Enable TCCEA are asserted.

A capture event occurs together with the assertion of MRL or TDL either on T2 or T4 positive edges, and captures the time-base value that caused the match, even if TCR1/2 increments concurrently with the assertion (see [Microcycle and I/O timing](#))<sup>27</sup>. MRLA/B and TDLA may, depending on the channel mode, inhibit the capture of the second event's TCR into CaptureA/B. As a general rule, values captured by signal transitions are not overwritten by values captured by match events.

When the enable bit TCCEA is asserted, captures due to Transition Events also occur after TDLA is asserted. Those captures happen on transition events specified by IPACA, and the TCR value is saved into CaptureA register only.

The capturing scheme is defined by the Channel Mode programmed at register PDCM, or at register UDCM when User Defined Channel Mode is selected. For more information on mode-dependent capture schemes refer to [Channel Modes](#).

### 35.15.5.3.1 TDLA/B - Transition Detect Latches

TDLA/B indicate detection of specific transition occurrences on a channel input signal. TDLA and TDLB assertion causes service request in single and double transition predefined modes, respectively. TDLB assertion does not cause Service Request in single transition predefined modes, and TDLA assertion does not cause Service request in double transition predefined modes. In single transition channel predefined modes TDLB can be asserted on the second transition, but does not generate Service Request. Yet on predefined modes, TDLB assertion is enabled only if TDLA is asserted to detect an ordered input signal double transition. All the restrictions above, however, may be overridden by using the User Defined Channel Mode. The IPACA and IPACB registers indicate the programmed edges of the first and second detected transition, respectively.

The sampling of a determined value (0 or 1) on the input signal due the occurrence of a Match is also treated as a "transition", depending on IPACA/B programming (see [IPACA/B and OPACA/B - Input and Output Pin Action Control Registers](#)). When using a channel mode where the transition1 is initially blocked and IPACA is programmed to detect such "transitions", the occurrence of a Match A only unblocks the transition after

---

27. In TPU3, when TCR1 was counting at maximum rate of eTPU clock divided by 2, the next value was captured.

the sampling. It means that the transition on the first Match (IPACA configurations 100 and 101) is not effective on predefined modes where Transition A is enabled by Match A (m2\_st, m2\_dt, m2\_o\_st and m2\_o\_dt) or user-defined modes with UDCM bit M1ET asserted. A Transition A can still happen after the Match A, however, if MatchA register is reprogrammed without clearing MRLA.

TDLA/B assertion conditions initiates a capture event of one or both selected TCR buses. TDLA or TDLB transition event generates a Service Request, depending on channel mode, previous events and the state of SRI. For more information on service request scheme refer to [Entry Point address generation](#) and [Channel Modes](#).

Assertion of TDLA/B occurs on either T2 or T4 positive edges. The capture event occurs on the same clock, and captures the time base value present when TDLA/B was asserted<sup>28</sup>. TDLA and TDLB are negated during reset and may also be negated independently by microcode. TDLA/B is reset by no way other than reset and microcode.

It is the transition from 0 to 1 in TDL that causes the Transition actions: even if TDL assert conditions are satisfied, no action due to a Transition occurs if TDL was already set to 1. However, if a Transition and a microoperation negating TDLs occur at the same time and TDL was already negated, TDL negation by microcode overrides its assertion, but any dependable captures and pin action occurs anyway.

### **35.15.5.3.2 TCCEA - Transition Continuous Capture Enable**

TCCEA enables capture from transitions after the TDLA flag is set. TCCEA is negated on reset, so that a capture occurs only when TDLA asserts. TCCEA can be set and reset by microcode only, through the instruction field MTD (see [Disable Match and Transition Service Requests](#)). It can only be set together with inhibiting of the channel service requests (SRI = 1)<sup>29</sup>.

When TCCEA is asserted, the transition events specified by IPACA that occur after TDLA is set also cause captures, into the CaptureA register only. However, output actions related to transition events are still blocked by TDLA.

---

28. In TPU3, when TCR1 was counting at maximum rate of eTPU clock divided by 2, the next value was captured.

29. TCCEA provides compatibility with TPU when service request is inhibited.

### 35.15.5.4 Channel Modes

The Enhanced Channels support various modes of operation combining Match A/B recognition and transition detection events which set MRLA/B and TDLA/B. The channel mode is individually set for each channel by eTPU microcode, through the PDCM register (see [PDCM - Predefined Channel Mode](#)). The PDCM register selects among a set of 13 predefined channel modes, and also a user-defined channel mode.

The order in which events occur, combined with assigned channel mode, establish which following event detections are inhibited and/or enabled, as well as the actions taken: Time Base capture, flag setting (MRLA/B, TDLA/B), match disabling (MRLEA/B), output signal transition, and Service Request. Those channel mode characteristics are fixed in the predefined modes, but can be individually programmed in the user-defined channel mode.

A generic description of channel modes from the usage point of view can be found in [Channel Modes Overview](#). Each mode is named with a mnemonic acronym for terse reference. The individual programmed attributes of the user-programmable channel mode are also described.

The modes are used differently for input and output signals, as explained in [Predefined Channel Modes on Input Signal Processing](#) and [Channel Modes on Output Signal Generation](#). Modes also allow combining input processing and output generation in a single channel, as exemplified in [Combining Input and Output Signals](#).

[Channel Mode Logic and Event Flags](#) shows a structural definition of channel logic and its relation to channel modes. A dynamic, event-oriented view of each channel mode can be found in [Channel Mode Summary](#).

#### 35.15.5.4.1 Channel Mode Logic and Event Flags

[Figure 35-307](#) shows a more detailed diagram of channel mode logic. The logic shown here is not necessarily identical to the actual channel logic implementation, but is equivalent with respect to conditions for event blocking, enabling, capture, and service requests.

Signals MSR, TSR, MCAP, TCAP, M1ET, M1EM2, M1BM2, M2BM1, M2BT, T1BM1, T2BM1, TBM2 and T1ET2 are decoded from programmed channel mode PDCM in predefined modes, and come directly from the UDCM register when user-defined mode is selected:

- TSR (1 bit) defines Service Requests issued by Transitions, as shown in [Table 35-348](#).
- MSR (2 bits) defines Service Requests issued by Matches, as shown in [Table 35-347](#).

- TCAP (1 bit) defines time base captures caused by Transitions, as shown in [Table 35-348](#)
- MCAP (1bit) defines time base captures caused by Matches, as shown in [Table 35-349](#).
- M1ET, M1EM2, M1BM2, M2BM1, M2BT, T1BM1, T2BM1, TBM2, T1ET2 (1 bit each) define Match and Transition reciprocal blocking and enabling, as well as Transition ordering, as shown in [Table 35-350](#) and [Table 35-351](#).

[Table 35-352](#) shows the decoded values of those control signals for each predefined channel mode. The first column shows the mnemonic reference for the predefined channel modes described in the following sections.

Changing PDCM or the UDCM when user mode is selected may set or reset any of the channel flags, or issue captures and service requests, so it is advisable to switch channel modes only in a quiescent channel state, with channel flags MRLEA/B, TDLA/B, MRLA/B cleared. Furthermore, an event flag asserted in one mode may keep asserted after the mode programming change, even if the flag is impossible to be set in the new mode.

**Table 35-347. MSR signals - Match Service Requests**

| value | MSR                                             |
|-------|-------------------------------------------------|
| 00    | issue no Service Requests on Matches            |
| 01    | issue Service Request on Match B only           |
| 10    | issue Service Request on 2nd <sup>1</sup> Match |
| 11    | issue Service Request on both Matches           |

1. 2nd Match means the Match that happens after the 1st Match in time, either Match A or Match B.

**Table 35-348. TCAP and TSR signals - Transition Captures and Service Requests**

| value | TCAP                                                                                                                                                                                                                                                                                  | TSR                                                      |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------|
| 0     | 1st <sup>1</sup> Transition captures Time Bases corresponding to Transition A and Transition B <sup>2</sup><br><br>Transition A captures corresponding Time Base if it is the second transition;<br><br>Transition B captures corresponding Time Base if it is the second transition. | issue Service Request on the 1st <sup>1</sup> Transition |
| 1     | Transition A captures corresponding Time Base.<br>Transition B captures corresponding Time Base.                                                                                                                                                                                      | issue Service Request on the 2nd <sup>3</sup> Transition |

1. 1st Transition means the Transition that happens first in time, either Transition A or Transition B.
2. Match capture(s) never overwrites a Transition capture. Transition captures can always overwrite a Match capture.
3. 2nd Transition means the Transition that happens second in time, either Transition A or Transition B

**Table 35-349. MCAP signal - Match Capture**

| value | MCAP                                                                                  |
|-------|---------------------------------------------------------------------------------------|
| 0     | Match A captures corresponding Time Base;<br>Match B captures corresponding Time Base |
| 1     | either Match captures Time Bases corresponding to Match A and Match B <sup>1</sup>    |

1. Match capture(s) never overwrites a Transition capture. Transition captures can always overwrite a Match capture.

**Table 35-350. TBM2 signal - Transition Blocks Match B**

| value | TBM2                        |
|-------|-----------------------------|
| 0     | Transition A Blocks Match B |
| 1     | Transition B Blocks Match B |

**Table 35-351. M1ET, M1EM2, M1BM2, M2BM1, M2BT signals**

| signal                                         | active value meaning                                                                   |
|------------------------------------------------|----------------------------------------------------------------------------------------|
| M1ET<br>(Match A Enable Transitions)           | Transitions are initially blocked,<br>and Match A enables Transitions                  |
| M1EM2<br>(Match A Enables Match B)             | Match B is initially blocked <sup>1</sup> ,<br>and Match A enables Match B             |
| M1BM2 <sup>2</sup><br>(Match A Blocks Match B) | Match B is initially enabled <sup>1</sup> ,<br>and Match A blocks Match B <sup>3</sup> |
| M2BM1 <sup>2</sup> (Match B Blocks Match A)    | Match A is initially enabled, and Match B blocks Match A <sup>3</sup>                  |
| M2BT<br>(Match B Blocks Transitions)           | Match B blocks Transitions                                                             |
| T1BM1<br>(Transition A Blocks Match A)         | Transition A blocks Match A                                                            |
| T2BM1<br>(Transition B Blocks Match A)         | Transition B blocks Match A                                                            |
| T1ET2<br>(Transition A Enables Transition B)   | Transition B is initially blocked, and Transition A enables Transition B               |

1. The initial condition of M1EM2 prevails over M1BM2, while M1BM2 blocking prevails over M1EM2 enabling, so that Match B stays always blocked when both M1BM2 and M1EM2 are active. This combination is used in single-match modes (sm\_\*).
2. Blocking of one Match by the other is done through MRLEs.
3. Matches always block themselves by resetting their own MRLEs (Match A always blocks Match A, Match B always blocks Match B)

**Table 35-352. Predefined Channel Mode control signals decoding**

| Predefined Mode | M SR | MCAP | M1 ET | M1EM2 | M1BM2 | M2BM1 | M2 BT | T1BM1 | T2BM1 | TBM2 <sup>1</sup> | T1ET2 | TS R <sup>1</sup> | TCAP <sup>1</sup> |
|-----------------|------|------|-------|-------|-------|-------|-------|-------|-------|-------------------|-------|-------------------|-------------------|
| em_nb_st        | 11   | 0    | off   | off   | off   | off   | off   | On    | off   | 0                 | On    | 0                 | 0                 |

Table continues on the next page...

**Table 35-352. Predefined Channel Mode control signals decoding (continued)**

| Predefined Mode      | M SR | MCAP | M1 ET | M1EM2 | M1BM2 | M2BM1 | M2 BT | T1BM1 | T2BM1 | TBM2 <sup>1</sup> | T1ET2 | TS R <sup>1</sup> | TCAP <sup>1</sup> |
|----------------------|------|------|-------|-------|-------|-------|-------|-------|-------|-------------------|-------|-------------------|-------------------|
| em_nb_dt             | 11   | 0    | off   | off   | off   | off   | off   | On    | off   | 1                 | On    | 1                 | 1                 |
| em_b_st              | 11   | 1    | off   | off   | On    | On    | off   | On    | off   | 0                 | On    | 0                 | 0                 |
| em_b_dt              | 11   | 1    | off   | off   | On    | On    | off   | On    | off   | 1                 | On    | 1                 | 1                 |
| bm_st                | 10   | 0    | off   | off   | off   | off   | off   | On    | off   | 0                 | On    | 0                 | 0                 |
| bm_dt <sup>2</sup>   | 10   | 0    | off   | off   | off   | off   | off   | off   | On    | 1                 | On    | 1                 | 1                 |
| m2_st                | 01   | 0    | On    | off   | off   | On    | off   | On    | off   | 0                 | On    | 0                 | 0                 |
| m2_dt                | 01   | 0    | On    | off   | off   | On    | off   | On    | off   | 1                 | On    | 1                 | 1                 |
| m2_o_st              | 01   | 0    | On    | On    | off   | off   | On    | On    | off   | 0                 | On    | 0                 | 0                 |
| m2_o_dt              | 01   | 0    | On    | On    | off   | off   | On    | On    | off   | 1                 | On    | 1                 | 1                 |
| sm_st                | 11   | 1    | off   | On    | On    | On    | off   | On    | off   | 0                 | On    | 0                 | 0                 |
| sm_dt <sup>2</sup>   | 11   | 1    | off   | On    | On    | On    | off   | off   | On    | 1                 | On    | 1                 | 1                 |
| sm_st_e <sup>3</sup> | 11   | 0    | off   | On    | On    | On    | off   | On    | off   | 0                 | On    | 0                 | 0                 |

1. signals TSR, TCAP and TBM2 replace the signal DTM used in previous eTPU versions.
2. bm\_dt and sm\_dt are exceptions in the match blocking logic by transitions. See [Both Match Request Modes \(bm\\_st, bm\\_dt\)](#) and [Single Match Modes \(sm\\_st, sm\\_dt\)](#).
3. sm\_st\_e is an exception in the capture scheme. See [Single Match Enhanced Mode \(sm\\_st\\_e\)](#).

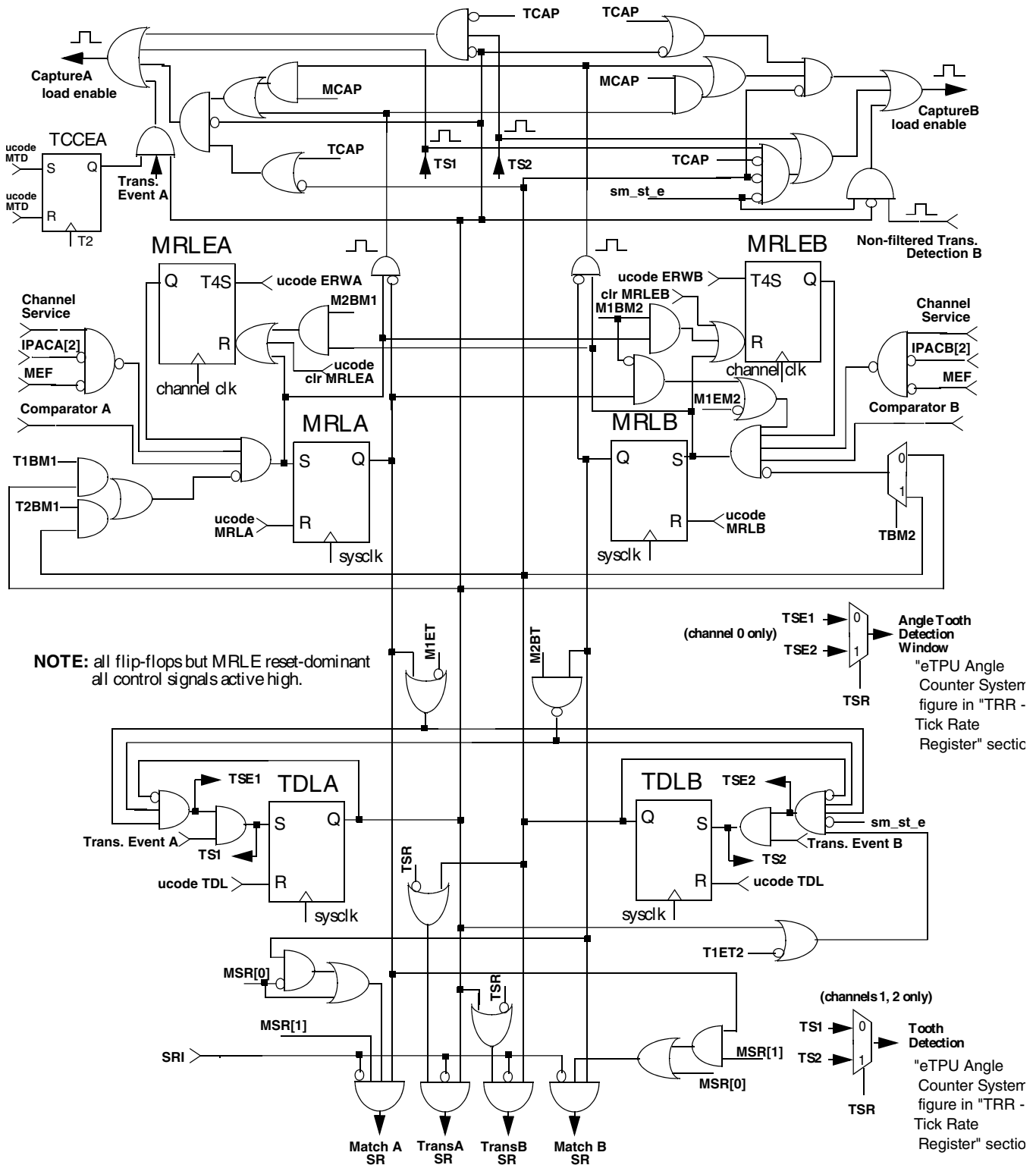


Figure 35-307. Channel Mode Logic and Event Flags

### 35.15.5.4.2 Channel Modes Overview

The predefined channel modes are divided, according to the way they treat transitions, in two basic modes:

- **Single Transition Modes** (mnemonic suffix **\_st**): in these modes the first transition (flagged in TDLA) issues a service request, and captures both time bases (selected by TBSA[1] and TBSB[1]) except on sm\_st\_e. The second transition (flagged in TDLB) does not issue a service request, but it captures time base selected by TBSB[1], except on sm\_st\_e.
- **Double Transition Modes** (mnemonic suffix **\_dt**): in these modes the second transition (flagged in TDLB) issues a service request, and each transition captures its own selected time base (Transition A and Transition B capture time bases selected by TBSA[1] and TBSB[1], respectively).

In predefined modes, Transition B is always (but not only) enabled by Transition A, so that transitions are always ordered: TDLA is set on the first transition and TDLB on the second. Unordered transitions are possible with user-defined mode, when UDCM bit T1ET2=0. Matches are generally not ordered, except on specific ordered match modes m2\_o\_st and m2\_o\_dt. Match capture(s) never overwrites a Transition capture, while Transition captures can always overwrite a Match capture, either in predefined or user-defined modes.

The following general rules apply to both predefined and user-defined modes:

- Blocking of one Match by the other, when it occurs, is done through MRLEs.
- Matches always block themselves by resetting their own MRLEs (Match A always blocks Match A, Match B always blocks Match B).

Predefined modes differ mostly by the way matches affects and are affected by other matches and transitions, as explained in next sections. However, some general rules on Match blocking apply:

- Match B is blocked by first transition (TDLA) in single transition modes, and by second transition (TDLB) in double transition modes.
- Both Matches are blocked by first transition in single transition modes.

#### Note

The rules above and in following sections may be overruled by the state of the channel latches if they are set/reset by microcode or if channel mode is changed. Care must be taken



to change channel modes, and is advisable to reset channel flags MRLA/B, TDLA/B and MRLEA/B before writing PDCM, or to UDCM when user-defined mode is selected.

#### 35.15.5.4.2.1 Either Match, Blocking Modes (em\_b\_st, em\_b\_dt)

In these modes the first match recognition that occurs blocks the other match recognition and generates a service request. They end up with one service request for two programmed match recognitions where only the first match recognition has an actual effect. If both match recognitions occur at the same time, both MRLA and MRLB are set, before the mutual blocking takes effect. The associated logic is shown in the following figure.

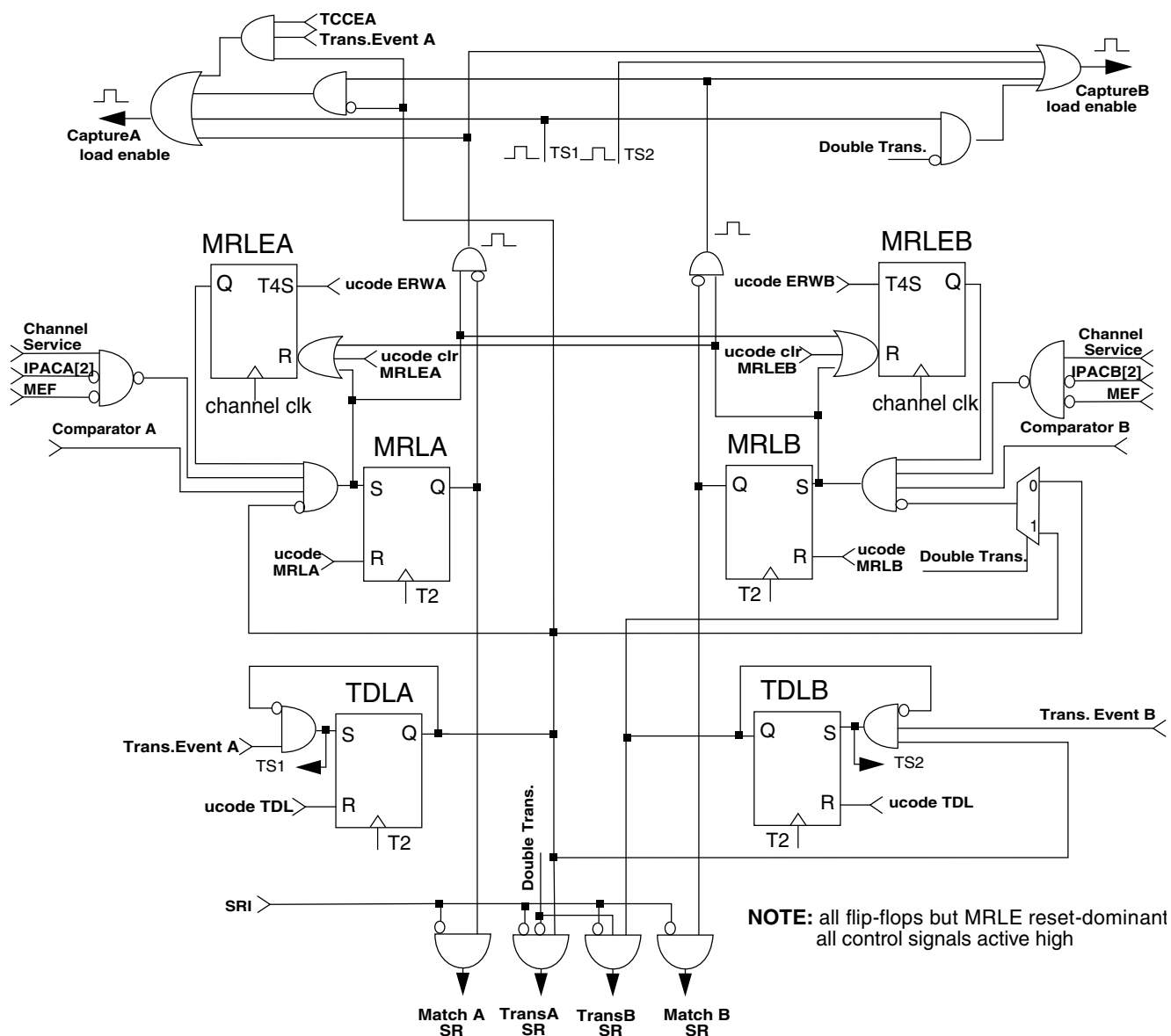


Figure 35-308. Either Match, Blocking Modes (em\_b\_st, em\_b\_dt)

### 35.15.5.4.2.2 Either Match, Non Blocking Modes (em\_nb\_st, em\_nb\_dt)

In these modes both match recognitions are independent and each of them generates service request. Each match recognition captures its related time base and does not block the other. The associated logic is shown in the following figure.

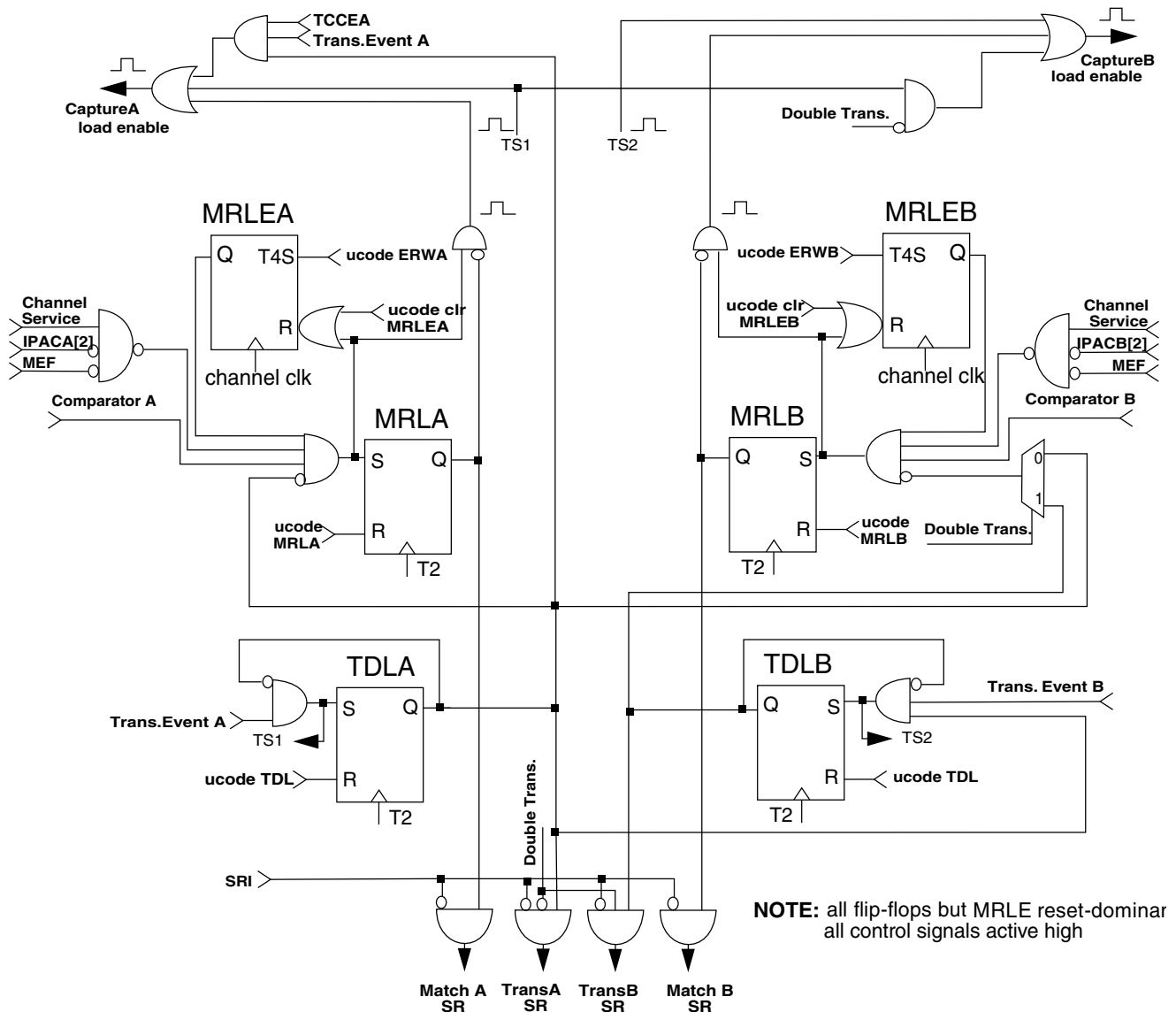


Figure 35-309. Either Match, Non Blocking Modes (em\_nb\_st, em\_nb\_dt)

### 35.15.5.4.2.3 Match B Request Modes (m2\_st, m2\_dt)

In these modes transitions are initially blocked, and are enabled by Match A. Match B recognition generates the match service request and disables Match A recognition. Each match recognition captures its own programmed timebase. In case of simultaneous match

recognition, both MRLA and MRLB are set, and OPACB register has priority over OPACA for selecting the pin action. The associated logic is shown in the following figure.

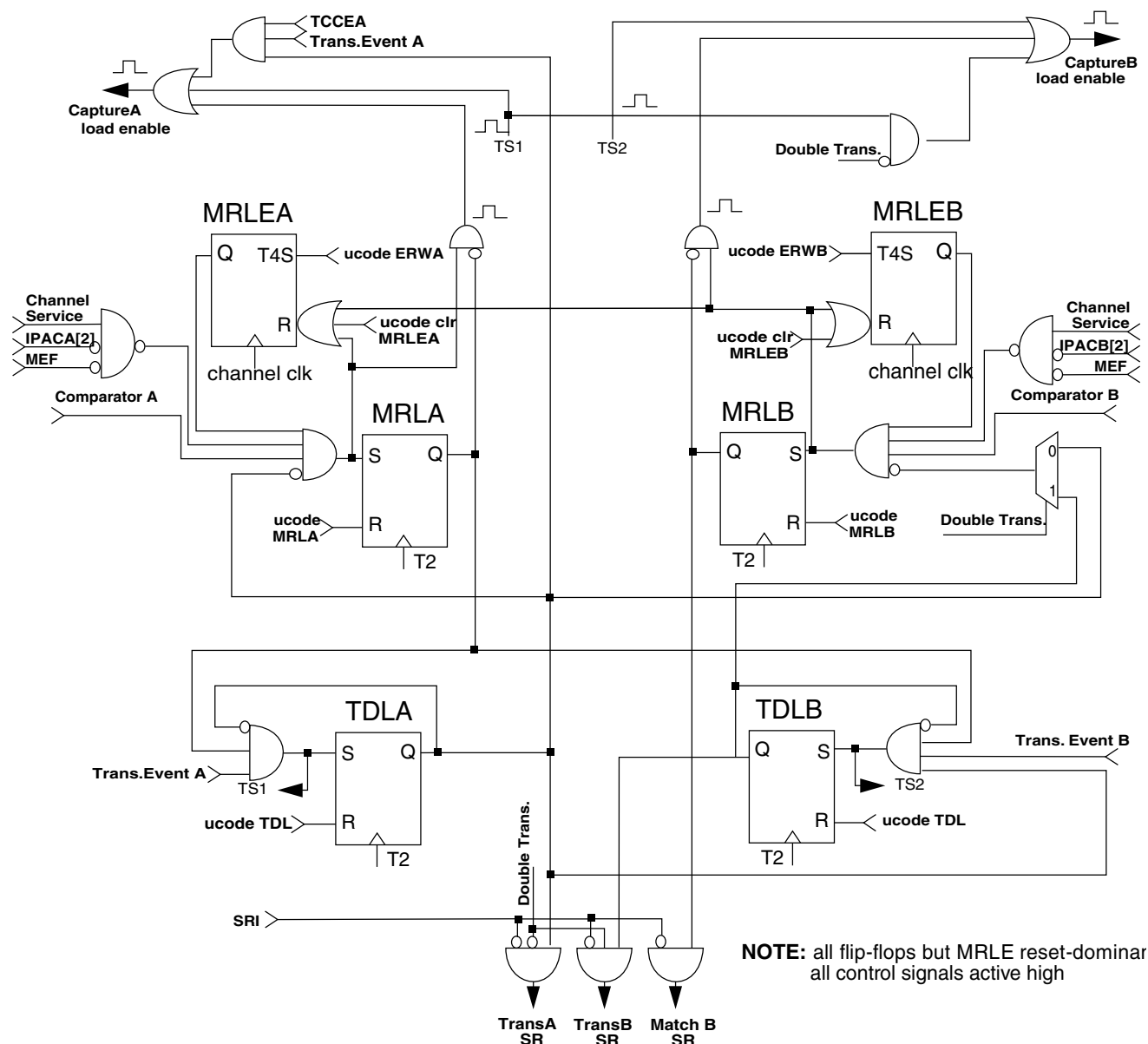


Figure 35-310. Match B Request Modes (m2\_st, m2\_dt)

#### 35.15.5.4.2.4 Both Match Request Modes (bm\_st, bm\_dt)

In these modes, match service request is generated only after both match recognitions occurred. By definition this is a non-blocking match mode: match recognitions do not block each other, implementing a last-served scheme. Unlike other double transition

modes, `bm_dt` blocks Match A with Transition B (not with Transition A), so that the second transition blocks both matches. The associated logic is shown in the following figure.

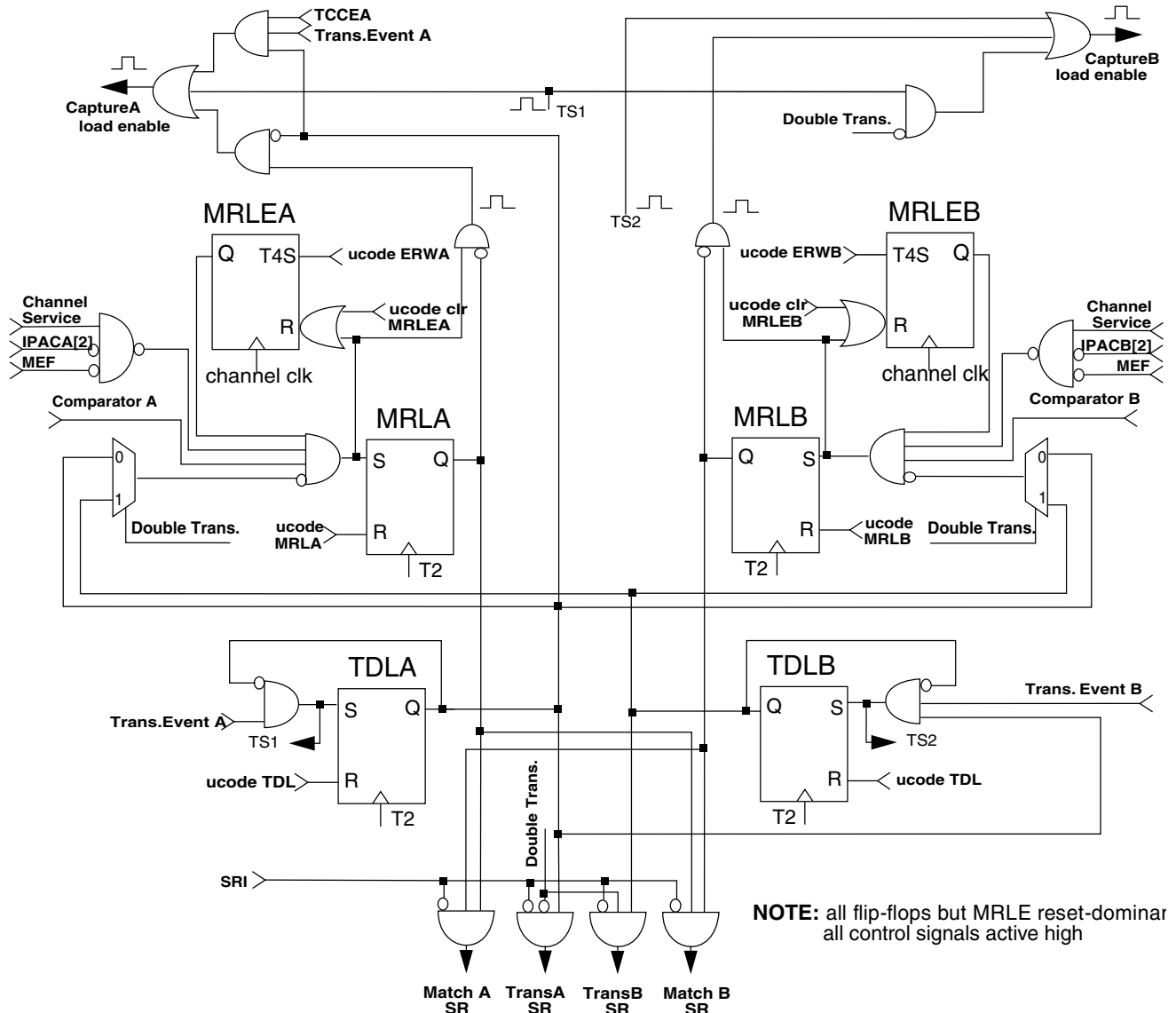


Figure 35-311. Both Match Request Modes (`bm_st`, `bm_dt`)

#### 35.15.5.4.2.5 Ordered Modes with Match B Request (`m2_o_st`, `m2_o_dt`)

These are ordered match modes on which Match A recognition must precede Match B recognition (ordered 1->2). Match A asserts MRLA and enables Match B and transitions. Match B asserts MRLB, generates a match service request, and blocks both transitions. The associated logic is shown in the following figure.

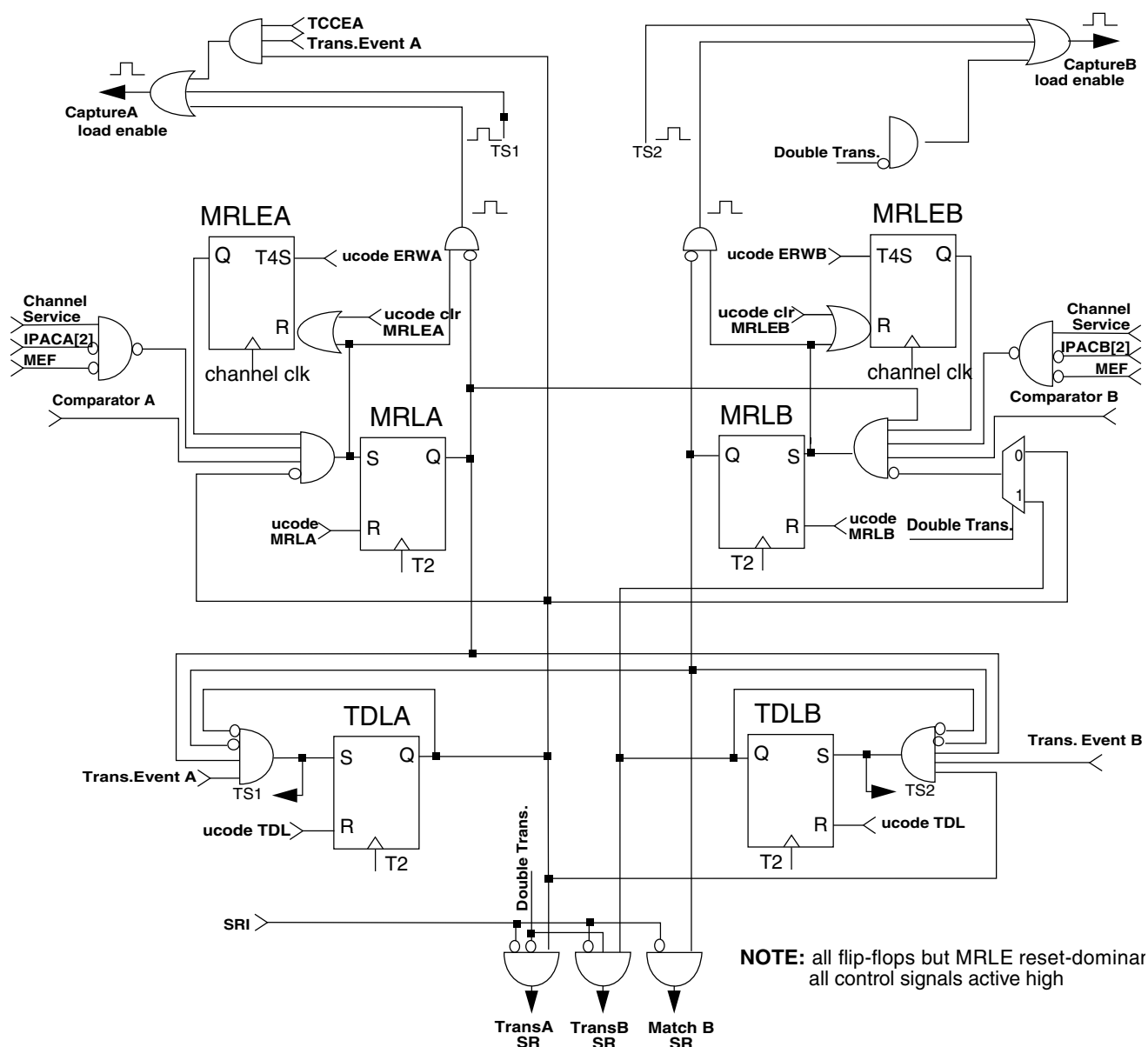


Figure 35-312. Ordered Modes with Match B Request (m2\_o\_st, m2\_o\_dt)

#### 35.15.5.4.2.6 Single Match Modes (sm\_st, sm\_dt)

Single match modes support single or double transition with single match recognition. MRLB is never set, and MRLEB has no effect. The associated logic is shown in the following figure.

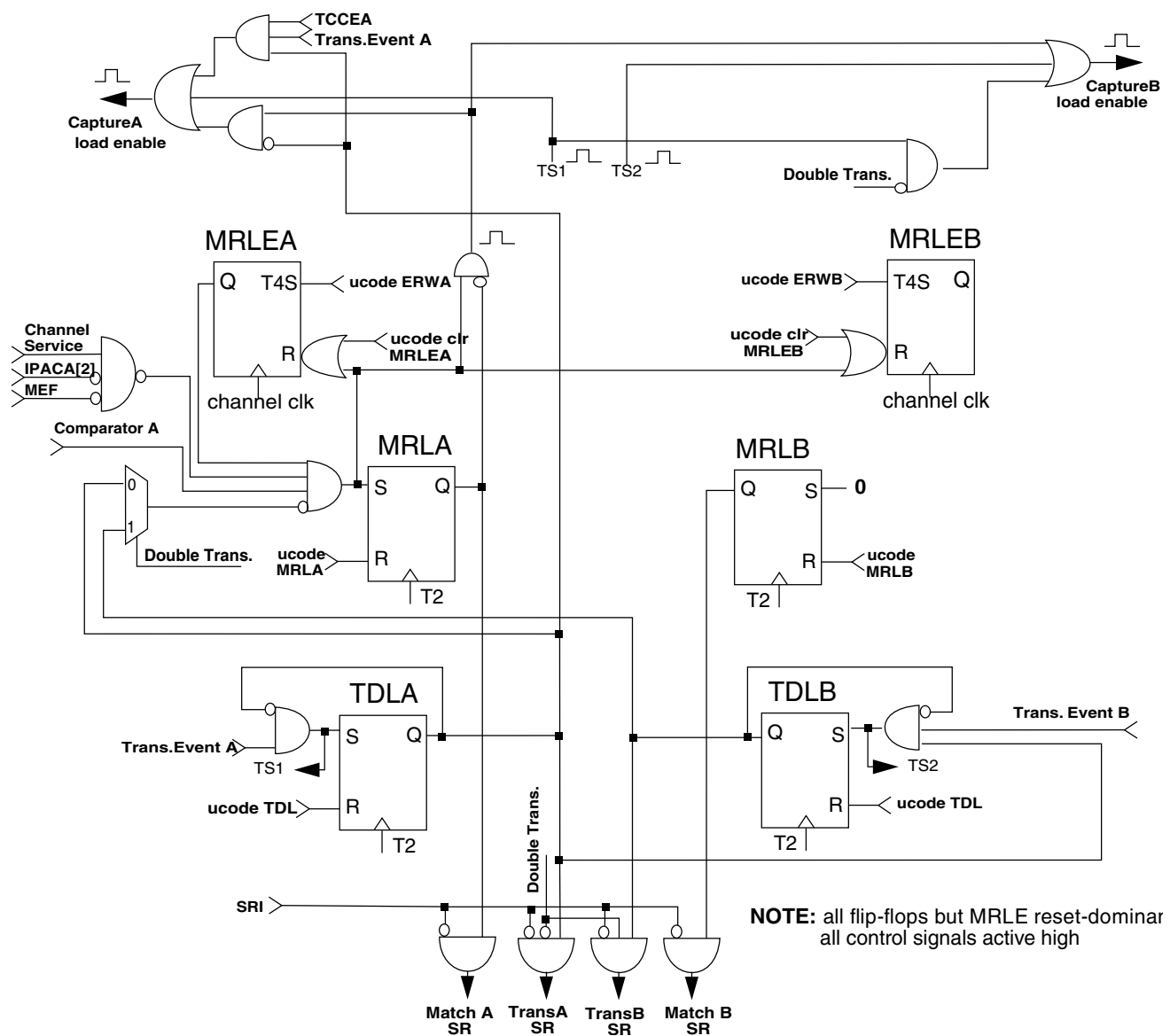


Figure 35-313. Single Match Modes (sm\_st, sm\_dt)

#### 35.15.5.4.2.7 Single Match Enhanced Mode (sm\_st\_e)

This is an enhanced single transition and single capture mode, which provides non-filtered input captures in addition to the single capture, allowing one to measure the delay of the digital filter. In an output channel, it has the same functionality of sm\_st (captures both time bases at once due to a match recognition). The associated logic is shown in the following figure.

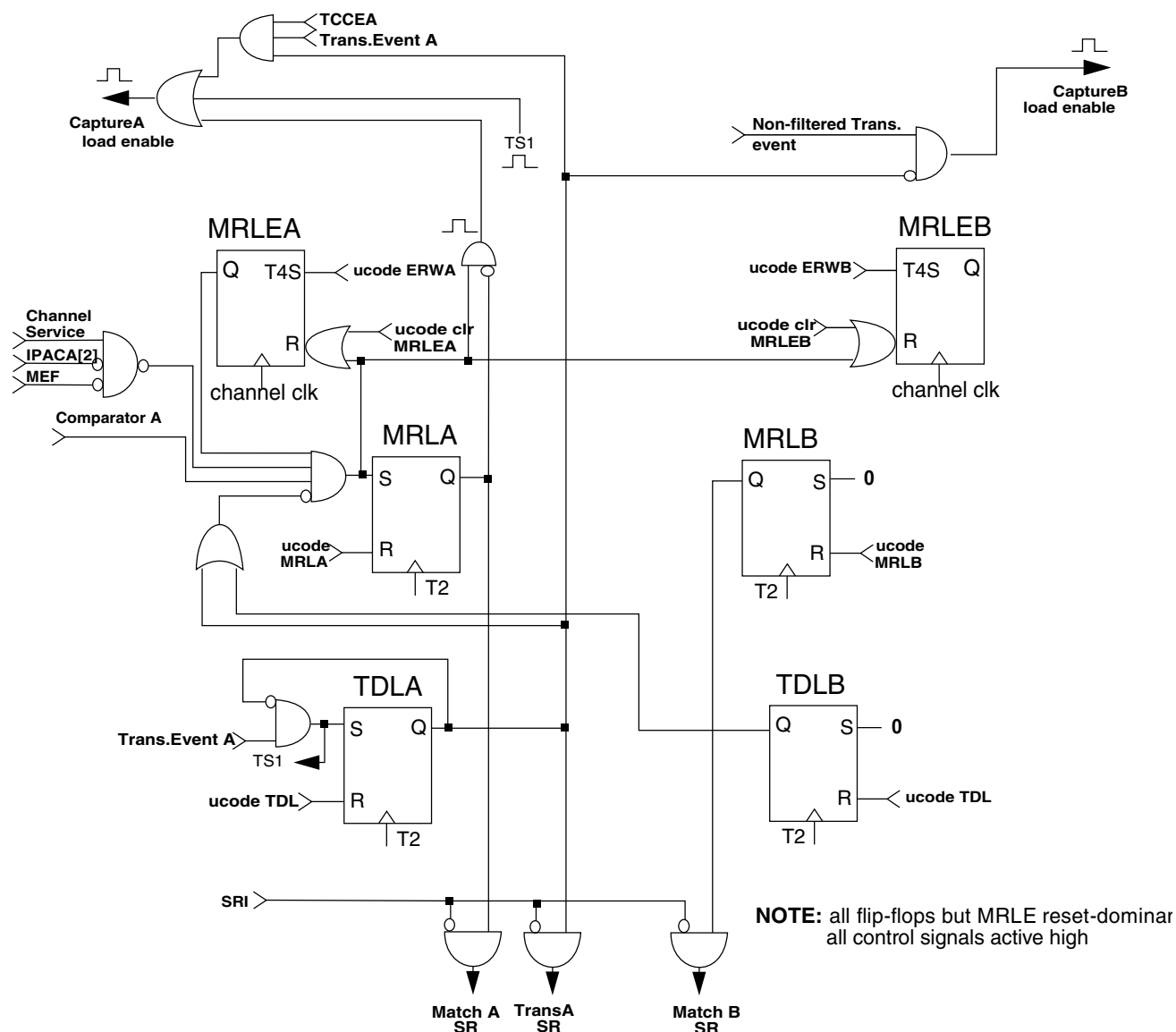


Figure 35-314. Single Match Enhanced Mode (sm\_st\_e)

### 35.15.5.4.3 Predefined Channel Modes on Input Signal Processing

When processing an input signal, the predefined channel modes can be classified in the following primary mode groups:

- Single Transition, Single Match: em\_b\_st, sm\_st, sm\_st\_e
- Single Transition, Double Match: em\_nb\_st, bm\_st, m2\_st, m2\_o\_st
- Double Transition, Single Match: em\_b\_dt, sm\_dt
- Double Transition, Double Match: em\_nb\_dt, bm\_dt, m2\_dt, m2\_o\_dt

In single transition modes, TDLA assertion may capture both time bases at once, while in double transition modes each transition captures its related time base in its related capture register.

Double transition is always ordered, i.e TDLB is enabled by TDLA and generates the service request.

The channel logic supports various input modes with combinations of single/double transition and single/double match, explained in the following subsections.

#### **35.15.5.4.3.1 Either Match, Blocking, Single Transition (em\_b\_st)**

On an input signal, this mode provides double timeout mechanism on a programmed transition edge with two timebases. The signal transition blocks both pending matches, indicating that no timeout condition occurred. The two match recognitions block each other, giving good separation in the entry table as to which match recognition caused the first timeout condition, and generating only one service request. Either match performs timebase captures which do not overwrite captures by transitions.

#### **35.15.5.4.3.2 Either Match, Blocking, Double Transition (em\_b\_dt)**

In double transition mode each transition is related to one match recognition. TDLA assertion captures its related timebase, blocks Match A and enables TDLB. TDLB assertion blocks Match B, captures its related timebase and generates a service request. Match recognitions block each other, so if there is a match timeout condition on TDLA, only one match service request is generated. This mode is good for qualifying two signal transitions by match timeout mechanisms, with one service request. Note that although TDLA assertion does not block Match B recognition, the value captured in CaptureA by TDLA assertion is not overwritten by this recognition. The second transition blocks Match B. Either match performs timebase captures which do not overwrite captures by transitions.

#### **35.15.5.4.3.3 Either Match, Non Blocking, Single Transition (em\_nb\_st)**

On an input signal, this is a double timeout mechanism of independent match recognitions of two different timebases. The match recognitions do not block each other, such that the microcode can check if one or two match recognitions occurred before their related signal transition. The signal transition detection (by IPACA) asserts TDLA, blocks both matches, captures both time bases and generates a transition service request, indicating that none of the two timeout conditions occurred. Any combination can be easily resolved by microcode (for example, signal transition after Match A and before Match B, or signal transition after both Match A and Match B).



Another possible use of this mode is allocating one match recognition for transition timeout and the other for another non-critical timed task, adding functionality to a single channel. Since the transition detection blocks both match recognitions, the match recognition of the other timed task is based on the fact that the comparator checks greater-equal conditions. It may be delayed if it occurs in the period between the signal transition detection (which blocks it) and the time TDLA is negated by microcode. If matches are enabled during the service, the same code can check if the match recognition of the timed task occurred in this period, by negating TDLA and writing to the CHAN register its own value (in order to update the MRLA flag in the branch logic).

#### **35.15.5.4.3.4 Either Match, Non Blocking, Double Transition (em\_nb\_dt)**

In this mode each transition is related to one match recognition, and the match recognitions are independent of each other. This mode can be used to give independent timeout conditions for the first and the second signal transition recognitions, and call service in any case of any timeout condition.

The first transition detection programmed in IPACA sets TDLA, captures its related timebase, blocks Match A recognition and enables TDLB assertion. The second transition detection programmed in IPACB sets TDLB, blocks Match B recognition, captures its related timebase and generates a service request. Any match recognition that occurs captures its related time base and generates a match service request, independent on the other match recognition.

#### **35.15.5.4.3.5 Match B Request, Single Transition (m2\_st)**

On an input signal, this mode provides an open window filter for a single signal transition. MRLA assertion opens the window, and enables transition detection on TDLA from this time on. MRLB assertion blocks Match A (by negating MRLEA), providing conditional window opening, because transitions are indirectly blocked. It also generates service request, but if it happens after Match A it does not block transitions, providing a non-blocking timeout mechanism for the estimated signal transition time (typically it indicates a missing transition, or mis-prediction of the transition time).

Transitions can be detected from the microcycle following MRLA assertion. The Transition A detection asserts TDLA, blocks both matches, captures both timebases and generates service request.

Using this mode, the channel can replace software open window filtering of qualified transitions with the channel hardware window. The window opening and timeout can be scheduled for any of the two time bases or combination of them. Typically, Match A will be used to open a prediction window, and Match B will be used as a timeout condition

which does not close the prediction window. This configuration improves noise immunity from early signal transitions, and reduces the probability for blocking late signal transitions due to timeout mis-prediction.

Using these conditions, the microcode can easily resolve the state:

- If TDLA and MRLA are asserted and MRLB negated, signal transition is in the expected range.
- If MRLA and MRLB are both asserted, and TDLA is asserted, the signal transition had a timeout condition due to Match B mis-prediction.
- If MRLB is asserted and TDLA negated, a timeout condition occurred, and the expected signal transition had not occurred yet.
- If MRLA is negated and MRLB is asserted, the conditional window did not open at all (for example: a time window is open only after a specific angle, otherwise it is not opened).

#### **35.15.5.4.3.6 Match B Request, Double Transition (m2\_dt)**

This mode is used as an open window filter for two signal transitions. In this case the Match A recognition opens the window (unless Match B recognition occurred first), and Match B recognition blocks Match A and generates a match service request. It is similar to m2\_st, but in this case, it is the second transition that blocks Match B. MRLB assertion is a global timeout condition for the two pulses. Like m2\_st, MRLB can conditionally eliminate the window from opening.

Using the TDLA, TDLB, MRLA and MRLB conditions, the microcode can easily resolve the state, in a similar manner as m2\_st, with additional information on the second transition (TDLB).

#### **35.15.5.4.3.7 Both Match Request, Single Transition (bm\_st)**

On an input signal, this is a double timeout mechanism on two different time bases. Both match recognitions must occur before the signal transition to generate a match timeout service request. Assertion of TDLA blocks both Match A and Match B recognitions, and captures both time bases, indicating there was no double timeout condition from both time bases.

Using the same timebase implements two timeout conditions, the first only sets its related MRL and the second generates a service request. Using these flags allows the microcode to check if one or both match recognitions precede the signal transition.

### 35.15.5.4.3.8 Both Match Request, Double Transition (bm\_dt)

In this mode the first transition detection does not block matches, since both match recognitions are required to generate a match service request. The second transition detection asserts TDLB, blocks Match A and Match B, captures its related timebase and generates transition service request. In this mode, a Match A recognition which occurs after the assertion of TDLA does not capture a new value in CaptureA, to preserve the actual signal transition time. Assertion of TDLA, however, always captures its related timebase.

This mode allows putting a double match timeout condition on the second transition. Typically, a pulse trailing edge timing can be checked against two time bases, to indicate if the pulse has not ended when both MRLA and MRLB are asserted. When a transition service request is generated by TDLB assertion, the state of MRLA and MRLB indicates which timeout condition occurred, if any.

### 35.15.5.4.3.9 Ordered Mode with Match B Request, Single Transition (m2\_o\_st)

On an input channel, this mode provides a closing window filter for a single signal transition. Match A assertion captures its programmed time base in CaptureA, opens the filter window (enables assertion of TDLA), and enables assertion of MRLB. Match B recognition captures its related timebase, closes the window (disables assertion of TDLA) and generates a service request. Due to Match A and Match B ordering, the window is opened for at least one microcycle. Match B recognition indicates a window timeout condition which blocks late signal transitions, outside the prediction window. Transition detection blocks both matches, indicating the transition occurred inside the estimated window. Transitions can be detected from the microcycle following MRLA assertion until the microcycle on which MRLB is asserted. When TDLA is asserted inside the window range it disables both matches, captures both time bases and generates a transition service request.

Using this mode, the channel can replace software window filtering of qualified transitions with the channel hardware window. The window opening and closing time can be scheduled for any of the two time bases or a combination of them.

### 35.15.5.4.3.10 Ordered Mode with Match B Request, Double Transition (m2\_o\_dt)

In this mode the channel logic implements a window filter for two detected signal transitions. MRLA assertion captures its related timebase and enables the assertion of both TDLA and TDLB. MRLB assertion captures its related timebase and disables the assertion of both TDLA and TDLB. Transitions can be detected from the microcycle following MRLA assertion until the microcycle on which MRLB is asserted. The first

signal transition (following MRLA assertion) asserts TDLA, captures its related timebase and enables assertion of TDLB. The second signal transition detection asserts TDLB, blocks Match B, captures its related timebase and generates the service request.

If both signal transitions occur inside the scheduled window, Match B recognition is blocked. If one or both signal transitions do not occur inside the scheduled window, Match B recognition generates a match service request and blocks further transition detections. The microcode can resolve the state using MRLA, MRLB, TDLA and TDLB, which affect the entry point selection.

#### 35.15.5.4.3.11 Single Match Enhanced Mode (sm\_st\_e)

This is an enhanced single transition and single match channel mode which provides timing information of the digital filter delay.

The CaptureA register captures the timebase selected by TBSA due to transition detection specified by IPACA or match recognition, as in sm\_st mode. Initially, the CaptureB register continuously captures the unfiltered IPACB-selected signal transitions from the digital filter input, directly from the signal synchronizer. When an IPACA-qualified, filtered transition detection occurs, TDLA is set, MRLA assertion is blocked, and, in addition, captures into CaptureB are also blocked. On service, CaptureA and CaptureB (copied into ERTA and ERTB) holds the time of the qualified transition detection (ERTA), and the time of the last signal transition at the input of the digital filter (ERTB). Subtracting the time in ERTB from the time in ERTA provides the delay of the digital filter.

In a quiet environment, the two captures provide the accurate delay of the digital filter in granularity of two eTPU clocks. In a noisy environment, false transitions may be detected at the input of the digital filter due to the noise, and the delay measurement may be reduced, especially if IPACB selects both edge detection. The microcode can do sanity checks on this value to recognize noise effects (for example: calculated delay is less than the minimum delay of the digital filter).

#### Note

In Channel 0, if eTPU\_TBR\_TBCR\_ENG $n$ [AM]=01 (Angle Mode), the unfiltered input comes from TCRCLK input and the filtered input comes from the TCRCLK filter output. The edge is selected by IPACA/B, and is independent of the edge selection by eTPU\_TBR\_TBCR\_ENG $n$ [TCR2CTL].

### 35.15.5.4.3.12 Single Match, Single Transition (sm\_st)

In this mode the channel logic is functionally back-compatible to a TPU3 single action channel, but a match or transition detection captures at once both timebases. The mode recognizes a single transition with single match timeout. Either TDLA or MRLA generates service request and captures both timebases. Assertion of TDLA blocks future assertions of MRLA.

### 35.15.5.4.3.13 Single Match, Double Transition (sm\_dt)

In this mode, the first transition detection asserts TDLA, captures a timebase in CaptureA and enables TDLB. The second signal transition asserts TDLB, blocks Match A, captures a timebase in CaptureB and generates a service request.

Match A (before TDLB) captures into CaptureB the timebase selected by TBSA, in order not to overwrite the captured value of TDLA.

This mode is used for scheduling one timeout condition on two input signal transitions (pulse timeout).

## 35.15.5.4.4 Channel Modes on Output Signal Generation

Since channel logic can generate output signal transitions based on Matches, the channel can be viewed as working in the following primary mode groups for signal generation:

- Single Match: em\_b\_st, sm\_st, sm\_st\_e, em\_b\_dt, sm\_dt
- Double Match: em\_nb\_st, bm\_st, m2\_st, m2\_o\_st, em\_nb\_dt, bm\_dt, m2\_dt, m2\_o\_dt

The channel logic supports various match channel modes with single/double match, as explained in the following subsections.

### 35.15.5.4.4.1 Either Match, Blocking Modes (em\_b\_st, em\_b\_dt)

On an output signal these modes are useful when using two different time bases to set a required signal transition. The first match condition which is met sets a required pin action, captures both time bases, blocks any effects of the other recognition, and generates a service request. Because the first match recognition blocks the other, the microcode can get good separation in the function entry table as to which match caused the timeout first, and both time bases are captured, enabling the microcode to compare one timebase to the other at the moment of the match recognition. These modes can be used for:

- Scheduling a required pin action to the first match recognition of two different time bases.
- Canceling a programmed pin action scheduled on one time base by match on another timebase (as a consequence of [Table 35-353](#)). Microcode has to set the OPAC register of the canceling match to no-action and the OPAC register of the other match to the required pin action which may be blocked. If Match A is the canceling match, it blocks the pin action also in case of two matches at the same time, since it has priority in this case. If Match B is the canceling match, it does not block the pin action in case of two matches at the same time.

#### **35.15.5.4.4.2 Either Match, Non Blocking Modes (em\_nb\_st, em\_nb\_dt)**

On an output signal these modes are useful in combination with the ME bit set on the entry point, to define an interlaced operation. For example, each match recognition can set a pin action, and the second pin action is not sensitive to microcode latency (ME bit asserted). Example for usage is PWM interlaced function on which the latency is determined by the period and not the duty cycle.

Another possibility is using one match for pin actions and the other match for an unrelated timed task without pin action (double the functionality of a single channel).

#### **35.15.5.4.4.3 Match B Request Modes (m2\_st, m2\_dt)**

On an output signal, these modes can generate narrow pulses or do conditional pin actions. A conditional pin action means that the pin state is changed only if the match recognitions occurred in the correct order, since the Match B recognition which generates the service request also has priority over the pin action and blocks future Match A recognitions.

Setting OPACA to a desired pin action and OPACB to no-action, and using different time bases for Match A and Match B defines a conditional OPACA pin action which can be blocked by Match B recognition. For example, setting Match A on time and Match B on angle can limit the pin action to a maximum angle value.

When pulses are generated, the service is requested at the trailing edge of the pulse, after MRLB is asserted.

#### **35.15.5.4.4.4 Both Match Request Modes (bm\_st, bm\_dt)**

On an output signal, each match recognition can affect the pin state, and capture its programmed time base. This way the pin action can be programmed separately for both match recognitions. For example, both match recognitions can negate the signal, and

service request is generated after both conditions are met. This mechanism can set two conditions to do a required pin action, and the first recognition changes the signal, but service is called only after both conditions occur.

When using the same time base, these modes can generate narrow pulses in any required order. For example, in a PWM function, when duty cycle is below 50% the function can get service on the low time and program the pulse to the required duty cycle of the high time. When duty cycle is equal or above 50%, the function can get service on the high time and program a negative pulse with the width of the required low time. To switch between the two states the function can program once the same transition time to MatchA and Match B with a required pin action, and on the next service program double match for the new state.

Another usage is generating a required pin action on one programmed time and service request later on another time, after the second match recognition occurs, or capturing some timebase on one time and generating a required signal transition and service request later.

#### **35.15.5.4.4.5 Ordered Modes with Match B Request (m2\_o\_st, m2\_o\_dt)**

The order of the match recognitions imply that OPACA register programmed pin action always precede the OPACB register pin action. Setting OPACA to no-action, based on the greater-equal comparator, enables using Match A on one time base to delay the signal effect of Match B on the other time base. This method implements a conditional pulse extension or conditional delay on signal transition.

These modes can also be used for deferred pulse generation with microcode service request after its trailing edge (if Match A condition comes after Match B condition). Another option is having Match A recognition associated with output pin actions and Match B recognition for a timed microcode task which has to be scheduled at a programmed time which may be delayed by the Match A pin action.

#### **35.15.5.4.4.6 Single Match Modes (sm\_st, sm\_dt, sm\_st\_e)**

There is no difference between plain and enhanced Single Match modes on an output signal.

In this mode the channel logic is functionally back-compatible to a TPU3 single match output channel. Match A recognition generates service request and sets the pin state according to OPACA register. It captures at once the timebase selected by TBSA in CaptureA and the timebase selected by TBSB in CaptureB.

### 35.15.5.4.4.7 Match/Transition Pin Action Conflict Resolution

In output signals, matches and/or transitions automatically cause pin actions defined by the OPACA/B and/or IPACA/B channel control registers (see [Pin Control Registers](#)). Simultaneous matches/transitions may be associated with different, possibly contradictory, pin actions. These conflicts are resolved according to the table below.

If an OPACA/B=000 (no action) prevails over non-zero OPAC according to the table below, then if Match A/Transition A and Match B/Transition B occur simultaneously, no output pin action occurs, that is: a match on the action logic with OPAC=000 inhibits simultaneous actions of the other OPAC, if prevailing according to the following table. That also applies when output actions are caused by inputs (OPAC=1xx).

**Table 35-353. Simultaneous Match Pin Action Priority**

| Channel Mode        | Priority                                                       |
|---------------------|----------------------------------------------------------------|
| em_nb_st / em_nb_dt | OPACA                                                          |
| em_b_st / em_b_dt   | OPACA                                                          |
| bm_st / bm_dt       | OPACA                                                          |
| m2_st / m2_dt       | OPACB                                                          |
| m2_o_st / m2_o_dt   | in these modes there is no possibility of simultaneous matches |
| user-defined        | OPACB if M2BM1 = 1 and M1BM2 = 0, OPACA otherwise              |

### 35.15.5.4.5 Combining Input and Output Signals

The processing of input signal can be combined with output signal generation. A detected input transition can trigger an output signal edge, even without microcode intervention, by using OPAC options 1xx.

The channel set-up examples below show these two capabilities combined (see the figure below).

The **first example** implements a fast (no microcode intervention) short-circuit protection feedback mechanism for driving high-current output devices. The signal after the high-current driver feeds back to the channel input. The input signal is normally delayed from the output signal by the device turn-on delay. After the channel output turns on, the channel logic must check if the driver output (connected to the channel input) follows the driven value after the maximum device turn-on delay. If it does not, the driver output is probably shorted, and the channel output must be turned off immediately to avoid damaging the device.



Match A causes the output to be driven high (for simplicity the output and input signals are shown as positive logic). It also causes a transition A, because  $IPACA=100$  and the input is still low. Match B occurs after the expected driver delay, and causes a service request. If the output is shorted, a Transition B occurs on Match B because  $IPACB=100$ . This will cause the output to go low immediately, because  $OPACB=100$ .

In the **second example**, an output pulse is generated from an input transition without microcode intervention. Match A opens a window for transitions and also enables Match B. A rising edge on input sets output high. On Match B the window closes, and input signal is checked: if sampled high, the output resets; otherwise it stays high.

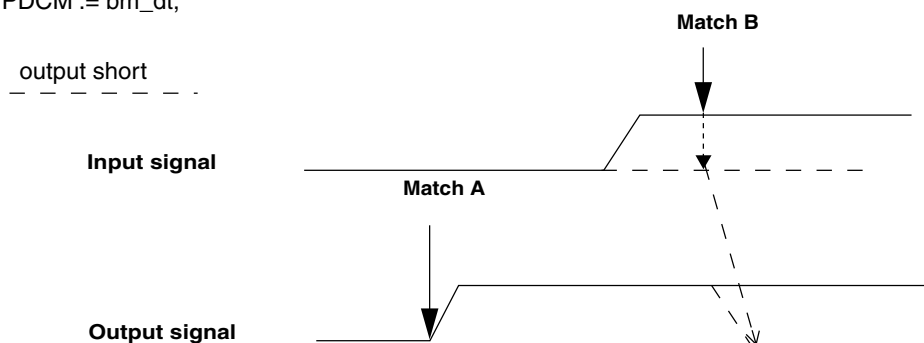
In the **third example**, a pulse is generated depending on the value sampled on the input signal at a predetermined time. Match A samples the input signal, causing a Transition. Low level on input sets output low, otherwise it stays high. Match B sets output high. In both cases a service request is issued (microcode intervention), at the beginning and at the end of the pulse (Match B), if required ( $SRI := 0$ ).

### Note

When  $IPAC = 1xx$ , a match event can cause simultaneously a Match recognition and a Transition detection. Depending on the Channel Mode, these Match and Transition may have conflicting effects on other Transition/Match blocking or enabling. In these cases, blocking always prevails over enabling, effective on the next microcycle.

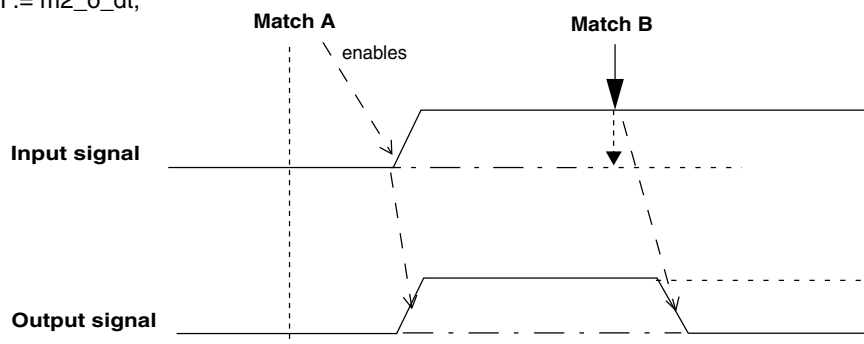
## Example 1: Short-circuit protection feedback

IPACA := 100; OPACA := 001; MatchA := output activate time;  
 IPACB := 100; OPACB := 100; MatchB := MatchA + max. high-current driver turn-on delay;  
 PDCM := bm\_dt;



## Example 2: Pulse generation on windowed input transition

IPACA := 001; OPACA := 101; MatchA := window open time;  
 IPACB := 101; OPACB := 100; MatchB := window close time, input sampling;  
 PDCM := m2\_o\_dt;



## Example 3: Pulse generation on input sampling

IPACA := 100; OPACA := 100; MatchA := window open time, input sampling;  
 IPACB := 000; OPACB := 001; MatchB := window close time = MatchA + pulse width;  
 PDCM := em\_nb\_dt;

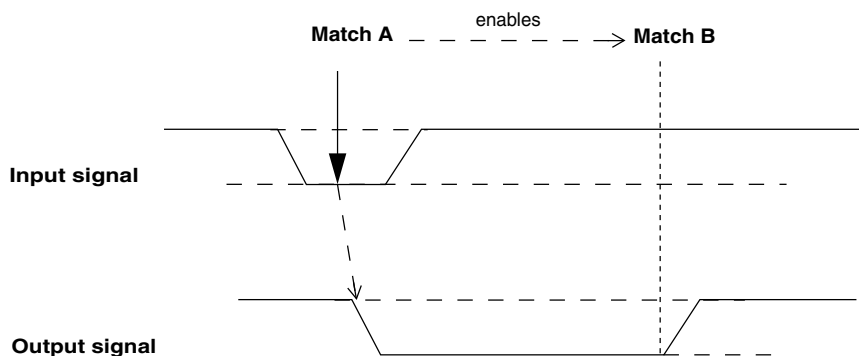


Figure 35-315. Input/Output combination

### 35.15.5.5 Channel Link

A channel can issue service requests to other channels through microcode, by assigning to the write-only microengine register LINK (refer to [LINK Register](#)) a value which specifies the target channel of the Link Service Request, as shown in the figure below.

Writing to the LINK register issues a link request to the target channel, setting its LSR flag. Each channel has its own LSR flag, which can be tested as a microcode branch condition (see [Conditional/unconditional branch](#)) and reset through the microcode field LSR (see [Clear Link Service Request](#)). The link branch condition samples, at the TST start, the value used to calculate the Entry Point.

Writing LINK with another channel target value in the same thread issues a Link Service Request to the new target, without negating the service request to the former one. This allows a channel to issue service requests to any number of channels, including itself. Neither LINK nor LSR microengine accesses are qualified by the CHAN register, i.e., they always access the serviced channel LINK and LSR, regardless of the value written in CHAN.

Channels can also receive link service requests from the Shared Subsystem Units (see [Shared Subsystems support](#)). These links are set by the interrupt signals from the shared subsystem and are level-sensitive, that is, they keep on requesting link service as long as the interrupt is active. Therefore, it is not enough to clear the link service request through the LSR instruction field: one must clear beforehand the respective shared subsystem unit interrupt source.

If microcode executes an instruction with field LSR=0 (clear Link Service Request), the link branch condition is cleared. However, the link service request itself is cleared only if no link from other channel was received by the serviced channel during the same thread<sup>30</sup>. Note that this applies only to links that come from the other engine or the same channel: if only Shared Subsystem link was received during the thread, the link service is also cleared with field LSR=0. If microengine clears LSR of its channel and, simultaneously, Link Service Request is issued to the current serviced channel, the branch condition is cleared but the link service request remains pending.

|                  |   |                       |                |   |   |   |   |
|------------------|---|-----------------------|----------------|---|---|---|---|
| 7                | 6 | 5                     | 4              | 3 | 2 | 1 | 0 |
| Engine Selection |   | Reserved <sup>1</sup> | Channel Number |   |   |   |   |

NOTES:  
<sup>1</sup> reserved bit must be written 0.

**Figure 35-316. Microengine LINK Register**

30. That can only happen if the link service request came from the other Engine, or from the serviced channel itself.

A channel can issue Link Service Requests to channels in any of two Engines, determined by the LINK register field Engine Selection (2 bits), as shown in the following table. In a Single Engine eTPU, Link is ignored when sent to the other engine, or engine 2.

**Table 35-354. LINK Engine Selection**

| Engine Selection | Description      |
|------------------|------------------|
| 00               | this Engine      |
| 01               | Engine 1         |
| 10 <sup>1</sup>  | Engine 2         |
| 11 <sup>1</sup>  | the other Engine |

1. ignored in Single Engine eTPU.

There is no explicit hardware mechanism to allow the thread to identify the link service request source, but it can be done by a software protocol using SDM parameters. All links are negated on reset.

### 35.15.5.6 Enhanced Digital Filter - EDF

The EDF eliminates passing of signal transitions which are caused by noise. Its purpose is to eliminate false transition service requests caused by noise pulses which are shorter than a programmed width.

See the chip-specific eTPU information for connections between eTPU and input glitch filters.

The EDF has three modes of operations, selected by the CDFC field in [eTPU Engine 1 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG1\)](#) and [eTPU Engine 2 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG2\)](#). These modes offer selections of trade-off between noise immunity and signal latency. CDFC also allows the filter to be bypassed. [Table 35-355](#) gives an example of minimum detected signal pulse and maximum filtered noise pulse in the three EDF operation modes. In Angle Mode, if AM=01, the EDF in channel 0 is replaced with the digital filter and synchronizer of the TCRCLK signal. In this mode, channel 0 works in combination with the Angle Counter logic, and their operation is fully synchronized.

Following subsections provide the functional description of the eTPU channel digital filter.

### 35.15.5.6.1 Two-Sample mode

In this mode the EDF works like the TPU2/3 digital filter. It uses the filter clock which is the eTPU clock divided by (2, 4, 8,..., 256) as a sampling clock. The filter clock is selected by the FPSCK field in [eTPU Engine 1 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG1\)](#) and [eTPU Engine 2 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG2\)](#). The EDF compares two consecutive samples. If both samples have the same value, the input signal state is updated. Note that when the FPSCK field selects the eTPU clock divided by two, the EDF works like the TPU1 four-clock digital filter.

### 35.15.5.6.2 Three-Sample mode

In this mode, like in the TPU2/3 mode, the EDF uses the filter clock as a sampling clock. The EDF compares three consecutive samples. If all three samples have the same value, the input signal state is updated.

The Three-Sample mode gives more signal latency than the Two-Sample mode, but also better noise immunity and better ratio between minimum detected signal pulse to maximum filtered noise pulse. When a certain filter clock frequency is selected for Two-sample mode, double filter clock frequency can be selected to get better latency in Three-sample mode.

### 35.15.5.6.3 Continuous mode

In this mode the EDF compares all the values sampled at the rate of eTPU clock divided by two, between two consecutive filter clock pulses. If the signal is continuously stable for the entire digital filter clock period (i.e all the samples have the same signal value), the input signal state is updated.

This method gives the same latency and the same ratio between minimum detected signal pulse to maximum filtered noise pulse, as the Two-Sample mode, as long as there is no noise. Each sampled noise delays the signal transition detection by at least a whole digital filter clock period.

The Continuous mode gives the best noise immunity by comparing multiple samples of the noise. On the other hand, when a short noise pulse appears in the middle of the filter clock period at the same time of a real signal transition, the Continuous mode may reject a real signal transition and delay the response to the first filter clock period in which the signal is continuously stable. This may add to the latency and also to the minimum detected signal pulse in a noisy environment.

### 35.15.5.6.4 Bypass Mode

In bypass mode the signal that feeds the edge detection comes directly from the output of the synchronizer, not filtered. Bypass mode automatically makes the channel logic work in T2/T4 timing mode (see [Channel timing modes](#)).

### 35.15.5.6.5 Filter clock prescaler

Each channel configured as an input has an associated synchronizer followed by a digital filter connected to the signal that samples signal transitions. After reset, the digital filter filters out high and low pulse widths smaller than the period of two eTPU clocks with eTPU\_SCR\_ECR\_ENG $n$  field FCSS=0, or 1 eTPU clock with FCSS=1 (see [eTPU Engine 1 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG1\)](#) and [eTPU Engine 2 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG2\)](#)), preventing these transitions from being input to the transition detect logic. For FPSCK=0 and FCSS=0, the synchronizer and digital filter are guaranteed to pass pulses that are as wide as or wider than four eTPU clocks, meaning a minimum period of 8 eTPU clocks. These figures are halved by setting FCSS=1. By changing the FPSCK field in [eTPU Engine 1 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG1\)](#) and [eTPU Engine 2 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG2\)](#) the user can select a lower clock rate for the filter signal to define wider valid pulses and filter out wider noise pulses. The filter prescaler clock control is a division of the eTPU clock. To guarantee pulse detection by the digital filter, the pulse must cover at least the stated number of samples at the filter clock rate. For example, a two sample digital filter must sample two points in the pulse to detect it. The table below shows the minimum guaranteed detected pulse width and the maximum filtered noise pulse width. The table refers only to the digital filter operation. The external pulses may have to be wider (to ensure detection) or narrower (to ensure filtering) depending on the rise/fall delay differences in the MCU receivers and internal logic. Delays introduced by synchronizer, filter and edge detection logic are explained in [Input/Output Signal Delays](#).

The TCRCLK signal also has a synchronizer and a filter. The FCSS bit has no influence on the clock prescaler of the TCRCLK filter, which works like the channel filter with FCSS=0.

**Table 35-355. Pulse widths and delays**

| Filter control (FPSCK) |          | Filter clock period <sup>1</sup> | Min. width guaranteed detected / max. width filtered<br>(min. filter delay / max. filter delay) <sup>1</sup> |                                              |
|------------------------|----------|----------------------------------|--------------------------------------------------------------------------------------------------------------|----------------------------------------------|
| FCSS = 0               | FCSS = 1 |                                  | Two-Sample or Continuous mode                                                                                | Three-Sample or Integrator <sup>2</sup> mode |
| not avail.             | 000      | 1                                | 2 / 1 (2 / 3)                                                                                                | 3 / 2 (3 / 4)                                |
| 000 <sup>3</sup>       | 001      | 2                                | 4 / 2 (3 / 3)                                                                                                | 6 / 4 (5 / 5)                                |

*Table continues on the next page...*

**Table 35-355. Pulse widths and delays (continued)**

| Filter control (FPSC) |            | Filter clock period <sup>1</sup> | Min. width guaranteed detected / max. width filtered<br>(min. filter delay / max. filter delay) <sup>1</sup> |                                              |
|-----------------------|------------|----------------------------------|--------------------------------------------------------------------------------------------------------------|----------------------------------------------|
| FCSS = 0              | FCSS = 1   |                                  | Two-Sample or Continuous mode                                                                                | Three-Sample or Integrator <sup>2</sup> mode |
| 001                   | 010        | 4                                | 8 / 4 (5 / 7)                                                                                                | 12 / 8 (9 / 11)                              |
| 010                   | 011        | 8                                | 16 / 8 (9 / 15)                                                                                              | 24 / 16 (17 / 23)                            |
| 011                   | 100        | 16                               | 32 / 16 (17 / 31)                                                                                            | 48 / 32 (33 / 47)                            |
| 100                   | 101        | 32                               | 64 / 32 (33 / 63)                                                                                            | 96 / 64 (65 / 95)                            |
| 101                   | 110        | 64                               | 128 / 64 (65 / 127)                                                                                          | 192 / 128 (129 / 191)                        |
| 110                   | 111        | 128                              | 256 / 128 (129 / 255)                                                                                        | 384 / 256 (257 / 383)                        |
| 111                   | not avail. | 256                              | 512 / 256 (257 / 511)                                                                                        | 768 / 512 (513 / 767)                        |

1. This table shows pulse widths and delays in number of periods of the eTPU clock.
2. Integrator mode is available for TCRCLK filtering only, see [TCRCLK digital filter](#).
3. These delays also apply to the TCRCLK filter when TCRCF selects eTPU clock/2 as filter clock.

### Note

The TCRCLK filter is clocked as if FCSS=0, always dividing eTPU clock /2 using FPSC, regardless of the FCSS field.

## 35.15.5.7 Channel timing modes

Channels can work on two different timing schemes, defining the period of channel clocking, tied to T2 and T4 microengine phases, as explained in subsections below. Microengine T2 and T4 phases are shown in [Microcycle and I/O timing](#).

### 35.15.5.7.1 T2 channel timing

In T2 timing mode the channel event state can only be updated every two eTPU clocks (see [Figure 35-346](#)). Pin state, TDLs, MRLs and Capture registers are updated on the microengine's T2 clock phase. MRLE clears also happen on T2, but MRLE setting occurs on T4, together with the Match register write by microcode (see [Write Channel Match and UDCM Registers](#)).

Channels work in T2 timing mode when either the T4DIS field in [eTPU Channel n Configuration Register for Engine 1 \(eTPU\\_CCCR\\_CnCR\\_ENG1\)](#) and [eTPU Channel n Configuration Register for Engine 2 \(eTPU\\_CCCR\\_CnCR\\_ENG2\)](#) is 1 or all the following conditions are true:

- The TCR1CS field in [eTPU Engine 1 Time Base Configuration Register \(eTPU\\_TBR\\_TBCR\\_ENG1\)](#) and [eTPU Engine 2 Time Base Configuration Register \(eTPU\\_TBR\\_TBCR\\_ENG2\)](#) is 0.
- The Enhanced Digital Filter is not configured as bypass (see [Enhanced Digital Filter - EDF](#)).
- The FCSS field in [eTPU Engine 1 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG1\)](#) and [eTPU Engine 2 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG2\)](#) is 0 (see [Filter clock prescaler](#)).

### 35.15.5.7.2 T2/T4 channel timing

In T2/T4 timing mode the channel event state can be updated on any eTPU clock (see [Figure 35-347](#)). Pin state, TDLs, MRLs, MRLEs, and Capture registers are updated either on microengine's T2 or T4 clock phases. MRLE clears can happen on T2 or T4, but MRLE setting occurs on T4 only, together with the Match register write by microcode (see [Write Channel Match and UDCM Registers](#)).

Channels work in T2/T4 timing mode when the T4DIS field in [eTPU Channel n Configuration Register for Engine 1 \(eTPU\\_CCCR\\_CnCR\\_ENG1\)](#) and [eTPU Channel n Configuration Register for Engine 2 \(eTPU\\_CCCR\\_CnCR\\_ENG2\)](#) is 0 and either one of the following conditions are true:

- The TCR1CS field in [eTPU Engine 1 Time Base Configuration Register \(eTPU\\_TBR\\_TBCR\\_ENG1\)](#) and [eTPU Engine 2 Time Base Configuration Register \(eTPU\\_TBR\\_TBCR\\_ENG2\)](#) is 1.
- The Enhanced Digital Filter is configured as bypass (see [Enhanced Digital Filter - EDF](#)).
- The FCSS field in [eTPU Engine 1 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG1\)](#) and [eTPU Engine 2 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG2\)](#) (see [Filter clock prescaler](#)).

T2/T4 timing mode demands special care on channel microcode that reprograms a match in a thread with matches enabled (entry point ME=1, see the “Entry Points” section). When the match register is written and the MRLA/B flag(s) is(are) cleared in the same instruction, a match can occur after MRLA/B is cleared, together with the set of respective MRLE. In this case, both MRL and MRLE will be set, and a match service request will occur if enabled. However, the match happened on the old match value, not on the new (reprogrammed) one. In order to prevent this ambiguity to the code that services the match, it is advisable to clear the MRLE (besides MRL) together with the



match reprogramming, avoiding the match on the old value to occur while the new match value is being written. The set of the MRLE due to match reprogramming prevails over the MRLE clear, thus allowing the new programmed match to occur in the future.

A similar case occurs when a match is blocked by a transition, the blocking TDL is cleared and the blocked match is reprogrammed at the same instruction: in T2/T4 timing mode: a match can occur on the old match value. The solution is the same as the case above, that is, clear the respective MRLE in the same instruction.

## 35.15.6 Time bases

Each eTPU engine has two Time Counter Registers, TCR1 and TCR2. They provide 24-bit time bases, shared by all 32 channels. Any channel can use both time bases to:

- Match channel's internal registers MatchA or MatchB;
- Capture time base value to channel's internal registers, CaptureA and/or CaptureB, when a match recognition or an Input transition detection occurs. For more information on channel events refer to [Enhanced Channels](#).

The TCR1 and TCR2 counters are accessible by the microcode for read and write operations. Its current value is used for getting the current time, and the captured values are used for channel relative time calculations of future events. When they are written at the same time they are incremented from any clock source, the written value prevails.

TCR1 with `eTPU_TBR_TBCR_ENGn[TCR1CS]=0` and TCR2 values are updated in T2 and read in T4 (see [Microcycle and I/O timing](#)). TCR1 can also work at full-speed eTPU clock, and so be updated on both T2 and T4, when `eTPU_TBR_TBCR_ENGn[TCR1CS]=1`. Both TCR1 and TCR2 values can be imported from or exported to the STAC bus. When their values are imported ( STAC Clients), these registers are written from the STAC bus and can only be read by microcode.

The TCR2 counters between the two Engines are out of phase by 1 eTPU clock, even when Time Bases are shared between them through STAC. It also applies to TCR1 counters if `eTPU_TBR_TBCR_ENGn[TCR1CS]=0`, but they can be in phase otherwise.

### 35.15.6.1 Timer Count Register 1 - TCR1

TCR1 can be used in the following modes:

- Internally Clocked Mode

## Functional description

- Externally Clocked Mode
- STAC Bus Client Mode

The host program can read TCR1 time base through [eTPU Engine 1 Time Base 1 \(TCR1\) Visibility Register \(eTPU\\_TBR\\_TB1R\\_ENG1\)](#) and [eTPU Engine 2 Time Base 1 \(TCR1\) Visibility Register \(eTPU\\_TBR\\_TB1R\\_ENG2\)](#).

The TCR1 bus runs through all the local engine channels. In channels which select TCR1 as MatchA and/or MatchB source, when its value is greater or equal to the programmed match value, a Match A and/or Match B event occurs on that channel. A recognized match event sets its related Match Recognition Latch 1 or 2, and according to the Predefined Channel Modes (PDCM) it may generate a channel service request. For details on eTPU channels see [Enhanced Channels](#).

### 35.15.6.1.1 Externally Clocked Mode

TCR1 can be driven externally by the TCRCLK input, after the digital filter. The TCR1 clock source is configured by the TCR1CTL bit, as shown in the following figure. For more information on clock source selection, see [eTPU Engine 1 Time Base Configuration Register \(eTPU\\_TBR\\_TBCR\\_ENG1\)](#) and [eTPU Engine 2 Time Base Configuration Register \(eTPU\\_TBR\\_TBCR\\_ENG2\)](#).

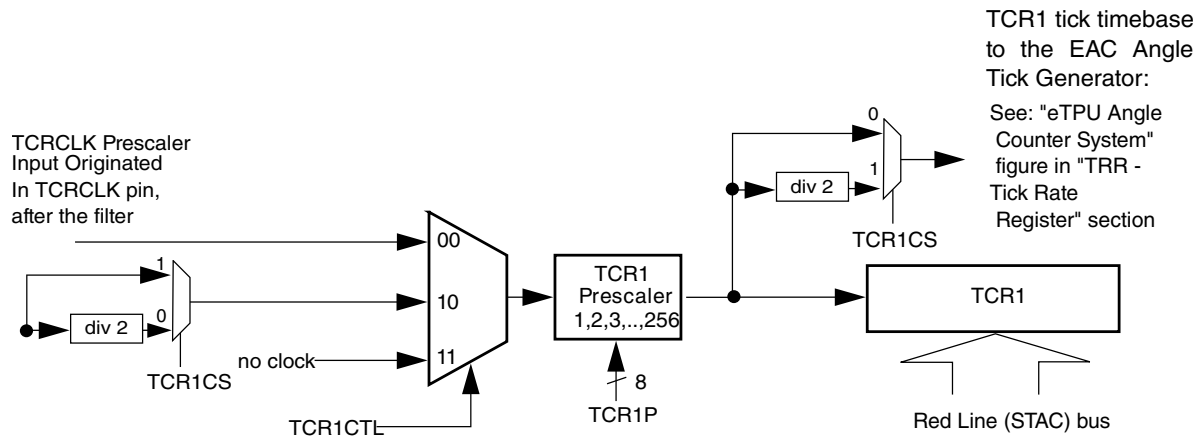


Figure 35-317. TCR1 clock selection

### 35.15.6.1.2 Internally Clocked Mode

TCR1 can be driven by the eTPU clock or eTPU clock divided by 2, before the prescaler. TCR1CTL can also be used to freeze TCR1 clock independently of TCR2 (unlike GTBE).

### 35.15.6.1.3 TCR1 clock prescaling

Any clock source selected by TCR1CTL is prescaled by a factor of 1 to 256, selected by the TCR1P field in [eTPU Engine 1 Time Base Configuration Register \(eTPU\\_TBR\\_TBCR\\_ENG1\)](#) and [eTPU Engine 2 Time Base Configuration Register \(eTPU\\_TBR\\_TBCR\\_ENG2\)](#). For more information on prescaler configuration see [eTPU Engine 1 Time Base Configuration Register \(eTPU\\_TBR\\_TBCR\\_ENG1\)](#) and [eTPU Engine 2 Time Base Configuration Register \(eTPU\\_TBR\\_TBCR\\_ENG2\)](#). The TCR1 Prescaler resets when the eTPU GTBE input is negated. After reset, it starts counting up to TCR1P when the eTPU GTBE input is asserted. When TCR1 increments (the eTPU GTBE input = 1), the prescaler starts a new count and the new TCR1P becomes effective. When TCR1 is written by microcode, the prescaler is reloaded with TCR1P and it becomes effective, if the eTPU GTBE input is asserted.

### 35.15.6.1.4 STAC Bus Client Mode

In this mode the TCR1 register is continuously updated from the STAC bus, and the clock selection and prescaling logic becomes ineffective. It is not writable by the microcode, and when read, it reflects the STAC bus imported value. The use of EAC is not allowed in client mode. This mode is configured through [eTPU Engine 1 STAC Configuration Register \(eTPU\\_TBR\\_STAC\\_ENG1\)](#) and [eTPU Engine 2 STAC Configuration Register \(eTPU\\_TBR\\_STAC\\_ENG2\)](#).

### 35.15.6.1.5 STAC Bus Server Mode

TCR1 bus can be exported to the STAC bus as a server, providing time information to other peripherals. This mode is configured through [eTPU Engine 1 STAC Configuration Register \(eTPU\\_TBR\\_STAC\\_ENG1\)](#) and [eTPU Engine 2 STAC Configuration Register \(eTPU\\_TBR\\_STAC\\_ENG2\)](#).

## 35.15.6.2 Timer Count Register 2 - TCR2

The TCR2 is a 24-bit counter which can be used in the following modes:

- Pin Transition Mode: Count the rise, fall or both transitions of TCRCLK signal.
- Angle Clock Mode: Count internal tooth angle in combination with the eTPU Angle Counter (EAC) hardware which implements an Angle PLL, and generates angle information to the channels. This mode is targeted for angle based applications.
- STAC (STAC) Bus Client Mode: TCR2 is driven by an external source (see [STAC Bus Client mode](#)).

- Gated Mode: Count with rate derived from the eTPU clock divided by eight. The TCRCLK signal is used to gate this count, enabling pulse accumulator operations.
- Internally Clocked Modes: TCR2 is driven by internal clock, with count rate of eTPU clock divided by eight.

All clock sources pass through a prescaler. In addition, the TCR2 count can be originated from the EAC which is a hardware angle clock and angle counter. Figure 35-318 shows the diagram for TCR2 clock control. When TCR2 is not driven by the EAC or STAC, the eTPU\_TBR\_TBCR\_ENGn[TCR2CTL] field selects the clock source, also allowing TCR2 to be frozen independently of TCR1 (see eTPU Engine 1 Time Base Configuration Register (eTPU\_TBR\_TBCR\_ENG1) and eTPU Engine 2 Time Base Configuration Register (eTPU\_TBR\_TBCR\_ENG2)). When in Angle Mode, TCR2CTL selects the TCRCLK edge sensitivity.

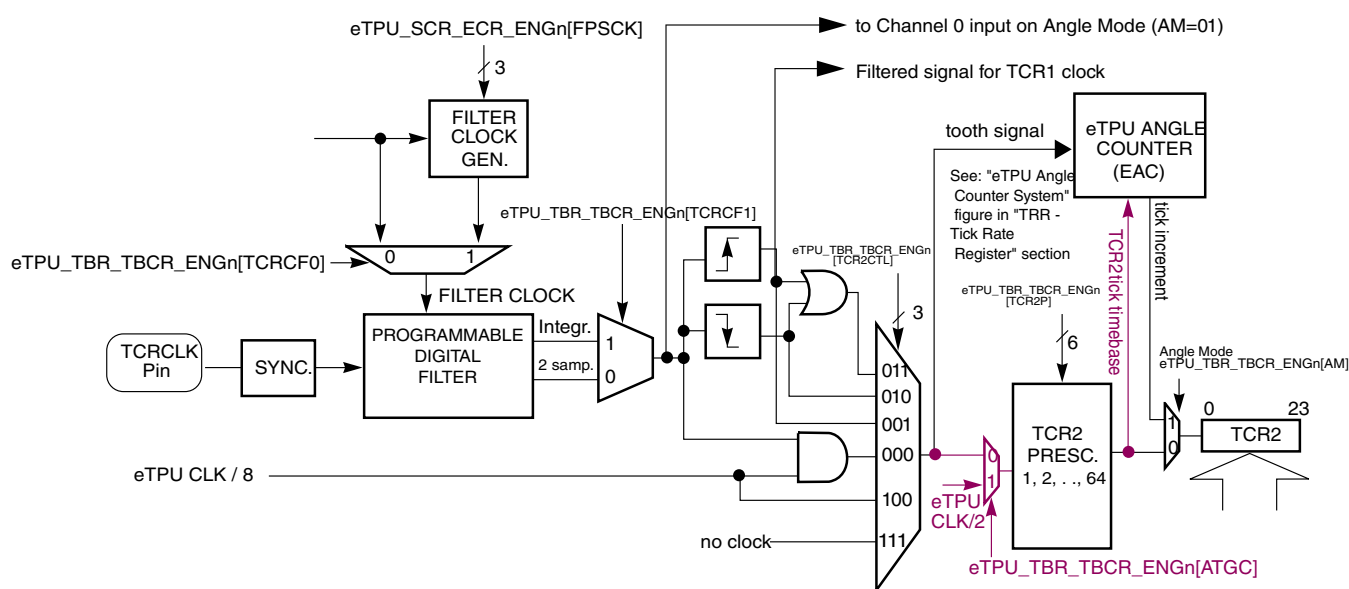


**Figure 35-318. TCR2 clock control**

The TCRCLK signal input is passed through a synchronizer and a programmable digital filter. In Angle Mode with AM=01, synchronizer and filter are also used in Channel 0, replacing its input synchronizer and filter, to get the same timing in the EAC and Channel 0. The TCRCLK synchronizer is an improved filter that provides best latency while maintaining proper noise filtering (see the field TCRCF in eTPU Engine 1 Time Base Configuration Register (eTPU\_TBR\_TBCR\_ENG1))

The TCR2 bus runs through all the local engine channels. It transitions on clock T2 (see Microcycle and I/O timing). In channels which select TCR2 as MatchA and/or MatchB source, TCR2 value is compared against MatchA and/or MatchB registers. A recognized

match event sets its related Match Recognition Latch 1 or 2, and according to the Predefined Channel Modes (PDCM) it may generate a channel service request. For details on eTPU channels see [Enhanced Channels](#).

The TCR2 counter is accessible by the microcode for read and write operations. Its current value is used for getting the current counter value (representing signal transitions, time or angle), and the captured values are used for channel relative count calculations of future events. The TCR2 value is readable to the host through [eTPU Engine 1 Time Base 2 \(TCR2\) Visibility Register \(eTPU\\_TBR\\_TB2R\\_ENG1\)](#) and [eTPU Engine 2 Time Base 2 \(TCR2\) Visibility Register \(eTPU\\_TBR\\_TB2R\\_ENG2\)](#). When the TCR2 bus value is imported from the STAC bus (STAC client mode), TCR2 is not writable by the microcode, and read access from the microcode or from the host reflect the imported TCR2 value.

### 35.15.6.2.1 TCR2 clock prescaling

Except in Angle Mode, any clock source selected by TCR2CTL is prescaled by a factor of 1 to 64, selected by `eTPU_TBR_TBCR_ENGn[TCR2P]`. For more information on prescaler configuration, see [eTPU Engine 1 Time Base Configuration Register \(eTPU\\_TBR\\_TBCR\\_ENG1\)](#) and [eTPU Engine 2 Time Base Configuration Register \(eTPU\\_TBR\\_TBCR\\_ENG2\)](#). The TCR2 Prescaler resets when the eTPU GTBE input is negated. After reset, it starts counting up to TCR2P when the eTPU GTBE input is asserted. When TCR2 increments (the eTPU GTBE input = 1), the prescaler starts a new count and the new TCR2P becomes effective. When TCR2 is written by microcode, the prescaler is reloaded with TCR2P and it becomes effective, if the eTPU GTBE input is asserted.

The counter that divides the eTPU clock by 8 before the prescaler also resets when the eTPU GTBE input is negated, or when TCR2 is written by microcode.

In Angle Mode the TCR2 prescaler is not used to directly increment TCR2, but can be used, selected by programming `eTPU_TBR_TBCR_ENGn[ATGC]=1`, as a clock source for the EAC generation of the angle ticks which increments TCR2. In this case, the TCR2 prescaler input is eTPU clock/2, independently of TCR2CTL. See [Generating the angle ticks](#) for details.

### 35.15.6.2.2 TCR2 Gated mode

TCR2 Gated mode is selected in the `eTPU_TBR_TBCR_ENGn[TCR2CTL]` field. In this mode the TCRCLK signal enables or disables transfer of the eTPU clock divided by 8 to the TCR2 prescaler. By programming the prescaler, TCR2 can run at rates from eTPU clock divided by eight down to eTPU clock divided by 512, in steps of eight eTPU clock

divisions. For more information, see [eTPU Engine 1 Time Base Configuration Register \(eTPU\\_TBR\\_TBCR\\_ENG1\)](#) and [eTPU Engine 2 Time Base Configuration Register \(eTPU\\_TBR\\_TBCR\\_ENG2\)](#).

### 35.15.6.2.3 TCR2 Signal Transition modes

These modes are selected when eTPU\_TBR\_TBCR\_ENG $n$ [TCR2CTL] is set to rise, fall or "rise-and-fall". In these modes the TCRCLK signal is the TCR2 clock source, and its maximum transition rate depends on the TCRCLK digital filter mode of operation. The TCRCLK digital filter can be programmed to use the eTPU clock divided by two, or use the same filter clock of the channels, controlled by eTPU\_TBR\_TBCR\_ENG $n$ [TCRCF]. It contains an up-down counter which operates as a digital integrator, optimizing signal latency in the selected mode and clock rate.

When eTPU clock divided by two is selected, the synchronizer and the digital filter are guaranteed to pass pulses that are wider than four eTPU clocks (two filter clocks). Otherwise the TCRCLK is filtered with the same filter clock as the channel input signals. For details on TCRCLK and channels digital filter control, see [eTPU Engine 1 Time Base Configuration Register \(eTPU\\_TBR\\_TBCR\\_ENG1\)](#), [eTPU Engine 2 Time Base Configuration Register \(eTPU\\_TBR\\_TBCR\\_ENG2\)](#), and [Enhanced Digital Filter - EDF](#).

### 35.15.6.2.4 STAC Bus Client mode

In this mode the TCR2 register is continuously updated from the STAC bus, and the clock selection and prescaling logic becomes ineffective. It is not write accessible for the microcode, and when read, it reflects the STAC bus imported value. The use of EAC is not allowed in client mode. This mode is configured through [eTPU Engine 1 STAC Configuration Register \(eTPU\\_TBR\\_STAC\\_ENG1\)](#) and [eTPU Engine 2 STAC Configuration Register \(eTPU\\_TBR\\_STAC\\_ENG2\)](#).

### 35.15.6.2.5 STAC Bus Server mode

When TCR2 bus is exported to the STAC bus as a server, it can provide either time or angle bus to other peripherals, according to its operation mode. This mode is configured through [eTPU Engine 1 STAC Configuration Register \(eTPU\\_TBR\\_STAC\\_ENG1\)](#) and [eTPU Engine 2 STAC Configuration Register \(eTPU\\_TBR\\_STAC\\_ENG2\)](#). To provide sequential update of the STAC clients, the Angle tick rate must not be faster than the STAC programmed update rate. This requirement puts a limitation on the angle clock count rate on high rate mode. In this case the Angle and Angle Fraction accumulator (see [Angle Tick Generator](#) and [Figure 35-324](#)) are advanced at rate of eTPU clock divided by eight. Therefore, the STAC update rate for the Angle Bus must not be slower than eight eTPU clocks.



### 35.15.6.2.6 TCR2 bus in Angle Clock mode

In this mode the TCR2 counter operates as part of the eTPU Angle Counter (EAC). The TCR2 bus value reflects this angle representation in which it counts Angle Ticks. Angle Mode is selected when `eTPU_TBR_TBCR_ENGn[AM]=1`.

Note that when TCR2 works in Angle Mode, it does not count directly from the TCR2 clock input which indicates tooth signal transition. Its Angle counter is controlled by the Count Control and High Rate logic (see [Count Control and High Rate Logic](#)), which provides the interpolated pin position, and handle cases of missing tooth, acceleration, de-acceleration and mechanical corrections.

The EAC uses the TCRCLK signal to get the tooth transition indications. The `eTPU_TBR_TBCR_ENGn[TCR2CTL]` field has to be set for the appropriate tooth edge detection rise, fall, "rise-and-fall" or none. TCR2 count clock comes from the EAC control and not directly from the physical tooth. This way the EAC control processes the signal transitions and handles missing teeth and flywheel mechanical corrections. Note that when TCR2CTL selects "none" for tooth edge selection, the TCR2 is not necessarily frozen, but can still be incremented by the EAC logic.

In Angle Mode, eTPU channel 0, 1 or 2 operation is combined with the EAC operation. When channel 0 is selected for EAC operation, the TCRCLK digital filter is used both by the EAC and by channel 0 to get full synchronization between the two logics.

The eTPU Angle Counter (EAC) logic runs continuously and updates the TCR2 Angle counter, eliminating the microcode latency in updating the TCR2 value.

### 35.15.6.3 STAC interface

Both time bases TCR1 and TCR2 can be shared between the Engines and with other blocks in the same MCU. Each one of both eTPU engines can drive their time bases to the STAC (Shared Time and Count) bus, acting as a server, while any other block can capture the value into its resources and behave like a client. For further reference about the STAC bus operation see [eTPU Engine 1 STAC Configuration Register \(eTPU\\_TBR\\_STAC\\_ENG1\)](#) and [eTPU Engine 2 STAC Configuration Register \(eTPU\\_TBR\\_STAC\\_ENG2\)](#).

The eTPU can export to the STAC bus or import from the STAC bus the following internal resources:

- TCR1: Can be exported to or imported from the STAC bus. TCR1 can only be imported from STAC bus when the engine is not in Angle Mode. When TCR1 is imported from the STAC bus, it becomes read-only for the microcode and reflects the imported values. For details, see [Timer Count Register 1 - TCR1](#).
- TCR2: Can be exported to or imported from the STAC bus. TCR2 can only be imported from the STAC bus when engine is not in Angle Mode. When TCR2 is imported from the STAC bus, it becomes read-only for the microcode, and reflects the imported values. When exported to the STAC bus, TCR2 can work in either Angle Mode or as a free running counter associated with the TCRCLK signal. For details, see [EAC - eTPU Angle Counter](#).

Proper configuration of the following bits is necessary to determine what can drive the STAC bus: eTPU\_TBR\_TBCR\_ENGn[AM] and eTPU\_TBR\_STAC\_ENGn[REN2, RSC2], according to the following table.

**Table 35-356. STAC bus and host read sources**

| AM<br>(eTPU_TBR_TBCR_ENGn) | REN2, RSC2<br>(eTPU_TBR_STAC_ENGn) | TCR2 bus source<br>(Host read of<br>eTPU_TBR_TB2R_ENGn) | STAC<br>bus driver |
|----------------------------|------------------------------------|---------------------------------------------------------|--------------------|
| 00                         | 0x (disabled)                      | TCR2/Time                                               | ×                  |
| 01, 10 or 11               | 0x (disabled)                      | TCR2/Angle                                              | ×                  |
| 00                         | 11 (Server)                        | TCR2/Time                                               | TCR2/Time          |
| 01, 10 or 11               | 11 (Server)                        | TCR2/Angle                                              | TCR2/Angle         |
| 01, 10 or 11               | 10 (Client)                        | not allowed <sup>1</sup>                                |                    |

1. STAC client configuration in Angle Mode is also not allowed for TCR1.

Note that Angle Mode is not available for STAC bus clients: configuring both at the same time brings unspecified results and must be avoided. When TCR2 is a stand-alone counter or a STAC Bus server, the same value that is driven to the internal TCR2 bus is also exported to the STAC bus (either Time Count or Angle).

STAC bus configuration is provided by the eTPU\_TBR\_STAC\_ENGn fields REN1/2 and RSC1/2. REN1/2 enable the STAC interface to interact with the resource (either TCR1 or TCR2 bus). RSC1/2 configure the resource (either TCR1 or TCR2 bus) as Server or Client.

Each time base / angle count resource from each engine receives a unique 4-bit hard-wired address that identifies it as a potential server. This address is used by the STAC Controller to coordinate which resource will drive the bus at a given STAC time-slot. For any time-slot there is a server driving the bus upon selection of the STAC Controller, and there may be a client linked to that server by the eTPU\_TBR\_STAC\_ENGn fields bits SRV1/2 on each Engine. When the server address on the Red Line STAC bus matches



the value in SRV1/2, the client will load the STAC information into the appropriate resource. For information on eTPU STAC Bus configuration, see [eTPU Engine 1 STAC Configuration Register \(eTPU\\_TBR\\_STAC\\_ENG1\)](#) and [eTPU Engine 2 STAC Configuration Register \(eTPU\\_TBR\\_STAC\\_ENG2\)](#).

The eTPU does not include a STAC Controller module, which is instantiated once in the system integration.

### Note

Setting a timebase as client of itself is not allowed.

#### 35.15.6.4 GTBE - Global Time Base Enable

**GTBE** enables time bases in both engines, allowing them to be started synchronously. GTBE is divided in two block interface signals: the eTPU GTBE output and the eTPU GTBE input. GTBE bit sets the eTPU GTBE output, and the eTPU GTBE input enables time bases to start. The eTPU GTBE output signal can be used by MCU integration for synchronization between eTPU time bases and time bases from other modules. If **GTBE** must enable only the eTPU time bases, the eTPU GTBE output is simply connected to the eTPU GTBE input. These two cases are shown in the figure below. Synchronization logic can be as simple as an OR logic gate.

Once the eTPU GTBE input transitions to 1, the Engine 1 Time Bases start 1 eTPU clock earlier than Time Bases in Engine 2, except when TCRCLK is selected as clock source or TCR1 when eTPU\_TBR\_TBCR\_ENG $n$ [TCR1CS]=1. This happens independently of prescaler values as long as they are the same for both engines, because the prescalers also freeze when the eTPU GTBE input = 0. Microcode can always write to TCR1/2 registers, with either value of the eTPU GTBE input.

### NOTE

The timebase prescalers are reset when the GTBE input is negated.

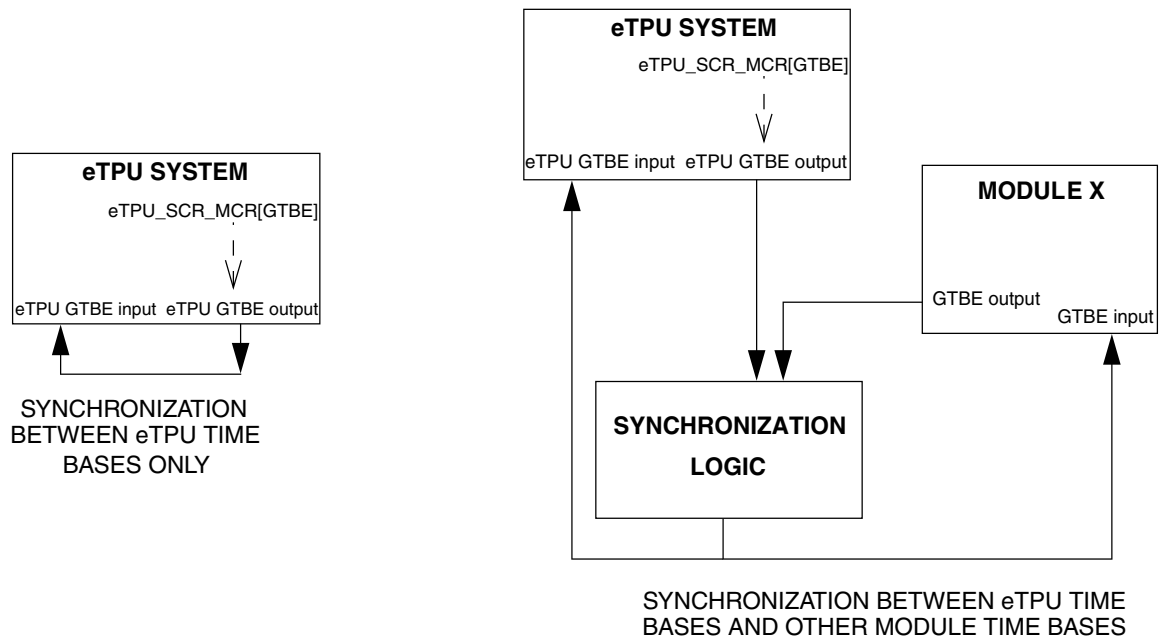


**Figure 35-319. Time base synchronization**

### 35.15.6.5 TCRCLK digital filter

The TCRCLK signal has an improved integrating digital filter with a 2-bit up-down counter. The counter counts up to 3 when a high signal level is detected, or down to 0 when a low level is detected. The signal state is updated to one when the counter stops at 3, or zero when the counter stops at 0. The `eTPU_TBR_TBCR_ENGn[TCRCF]` field (see [eTPU Engine 1 Time Base Configuration Register \(eTPU\\_TBR\\_TBCR\\_ENG1\)](#) and [eTPU Engine 2 Time Base Configuration Register \(eTPU\\_TBR\\_TBCR\\_ENG2\)](#)) determines whether the TCRCLK signal input (after a synchronizer) is filtered with the same filter clock as the channel input signals (see [Enhanced Digital Filter - EDF](#)) or uses the eTPU clock divided by 2, and also whether the TCRCLK digital filter works in integrator mode or the same two sample mode as the channel filters (see [Table 35-339](#)).

The TCRCLK filter delay and prescaling determines the minimum detectable TCRCLK pulse widths and, therefore, its maximum frequency, as shown in [Filter clock prescaler](#) and [Table 35-355](#). The TCRCLK signal delay from the module input to TCR1/TCR2 incrementing or detection in the EAC logic is explained in [Input/Output Signal Delays](#).

## 35.15.7 EAC - eTPU Angle Counter

### 35.15.7.1 General

The EAC logic contains a mechanism which follows the flywheel angle, based on the tooth rate. This hardware works in combination with the TCRCLK signal, the TCR2 counter and Channel 0, 1 or 2 (depending on the eTPU\_TBR\_TBCR\_ENG $n$ [AM] field) to generate angle information on the TCR2 bus which is passed to all the local engine channels. The EAC helps to implement a digital angle PLL (see [Figure 35-323](#)), which combines hardware with microcode processing at channel 0, 1 or 2. The angle measurement is based on history knowledge of the tooth period, for predicting the period of the next tooth. The tooth period is partitioned into a programmable number of **Angle Ticks**. The eTPU application will use the divider in the MAC/Divide unit to calculate an integer and a fraction part of the angle tick such that the full tooth period gets the correct programmed number of angle ticks with no accumulated error.

Each single tooth can be divided in angle ticks, up to 1024. In a 60 teeth flywheel, 128 Angle Ticks per tooth provide resolution of ~0.05 degrees per tick, which meets the accuracy requirement of 0.1 degrees in current automotive applications.

The measurement of one tooth in angle ticks is independent on engine RPM; it is the tooth period itself (and the corresponding tick period) that is re-calculated for each new tooth, based on the difference between the estimated tooth and the actual detection.

For these applications, one of the eTPU channels 0, 1 or 2 is dedicated to service the physical tooth detection. Channel 0 shares the same filtered input as the TCRCLK signal to get the same timing as the EAC. The TCRCLK edge detection is selected by the eTPU\_TBR\_TBCR\_ENG $n$ [TCR2CTL] field for the EAC, and by IPACA/B on channel 0, which must be set to detect the same edge(s). When channels 1 or 2 are selected to work with the EAC, IPACA/B is used to select the tooth signal edge detection for both the channel and the EAC, and the tooth signal that feeds the EAC is the same filtered input which feeds the channel.

Channel 0, 1 or 2 generates the signal transition service request, and can also be used for generation of a window filter on this transition, to qualify TCR2 clocks. For this purpose, the selected channel should be configured with double match window filtering mode (refer to [Channel Modes](#)). Depending on the channel mode set for the channel, Match A recognition opens the window, and Match B recognition may close it or leave it open. See [Angle Logic and Channel Modes](#) for details. Match B also generates a time-out service request. Its input signal transition comes from the tooth. The window can be defined by microcode to open at a predefined point inside the tooth period, and stay open for a desired percentage to the tooth period. The window can be measured in angle or time. This method improves the noise immunity by allowing transition detection only on an expected period, a feature which was software responsibility in previous TPU versions.

The EAC supports deceleration, acceleration, last tooth and missing tooth scenarios. The large range of angle ticks per tooth can be used to cover longer tick counts caused by one or more missing teeth, or to provide extra resolution for future application requirements. In case of a missing tooth, the EAC can be configured to insert a dummy tooth or to simply measure a longer tooth.

Figure 35-324 shows the block diagram of the Angle Counter system. TCR1 or TCR2 prescaler is used as a time base which measures the tooth period and is used for partitioning the period to angle ticks.

35.15.7.2 Angle Mode registers

In Angle Mode, the registers described below control eTPU angle operations. They are accessible only by microengine as source and destination registers in microinstructions. When eTPU is not in angle mode (eTPU\_TBR\_TBCR\_ENGn[AM] field is negated), all Angle Mode registers can be used as general purpose registers.

35.15.7.2.1 TPR - Tooth Program Register

TPR provides configuration for the Angle Counter circuit. In this register, the microcode can properly adjust the tooth count (controlling last tooth, missing teeth, dummy tooth insertion, halt until tooth detection) and the number of angle ticks per tooth (field TICKS). Note that this register is sampled into a temporary register in the EAC logic when the High Rate Mode is detected (see High Rate Mode (Acceleration) and TPR Buffering), which means that changes to this register may take effect only for the next tooth.

See Count Control and High Rate Logic and to Special Cases of Missing Teeth and Last Tooth until Handling False Tooth Detection for a detailed explanation about the use of this register. The figure below provides a detailed description of TPR register.

Several conflict issues on TPR writes are explained in Special TPR write cases.

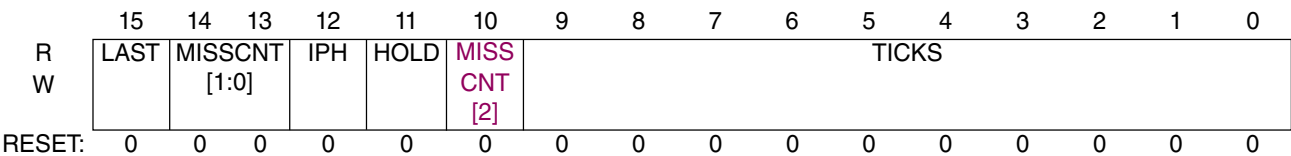


Figure 35-320. TPR

LAST —Last Tooth Indication  
Asserted by microcode and negated when a tooth is detected or inserted via IPH.

Table continues on the next page...

1 = Last Tooth - reset TCR2 Counter at the end of the tooth tick count (after physical tooth or IPH=1) when MISSCNT=0.

0 = Not Last Tooth.

If the tooth is detected or inserted before the tooth tick count completes (going High Rate mode, see [High Rate Mode \(Acceleration\)](#)), LAST resets immediately, but TCR2 resets only when the tooth count completes and MISSCNT=0.

#### MISSCNT — Missing Tooth Counter

Decrement on each estimated tooth, stops at zero. Used for generation of up to 7 "Dummy Teeth" whenever it holds a non-zero value.

If the tooth is detected or inserted before the missing tooth tick count completes (going High Rate mode, see [High Rate Mode \(Acceleration\)](#)), MISSCNT resets immediately, but missing teeth count continues in High Rate mode (see [TPR Buffering](#)).

#### IPH — Insert Physical Tooth

This bit generates a dummy physical tooth which has the same effect as a real physical tooth, and resets itself subsequently. If EAC is in Halt mode, it switches back to Normal mode<sup>1</sup>. If EAC is in Normal Mode, it switches to High Rate Mode. If Angle Logic is frozen by HOLD=1 (see below), it returns to the state it was at the freezing moment.

1 = Insert dummy physical tooth.

0 = No Operation.

**Note:** IPH reads as 1 in the next microinstruction after it is asserted, negating subsequently. However, it can be set twice in two consecutive microinstructions to generate two teeth and make the EAC go from Halt to Normal to High Rate Mode.

#### HOLD — Force EAC Hold

This bit forces the EAC to halt its operation in a special EAC freeze mode until a new physical tooth (a real one or emulated with IPH=1) is detected. Assertion of this bit immediately freezes the EAC in the middle of the tooth period. When a new physical tooth is detected, the bit is automatically negated by the EAC. The HOLD bit can be used for synchronizing the EAC tooth count, in case that a false physical tooth is detected due to noise.

1 = Force EAC to halt until detection of a physical tooth.

0 = Normal Operation.

#### TICKS — Angle Ticks Number in the Current Tooth

This field defines the number of angle ticks in the current physical tooth. It partitions the tooth period to the required number of angle ticks. The actual number of angle ticks in a tooth is (TICKS+1).

**Note:** In High Rate mode (see [High Rate Mode \(Acceleration\)](#)), TPR writes are immediately effective only for bits IPH and HOLD. All other fields changes are "buffered" and become effective when EAC leaves High Rate mode. See also [Special TPR write cases](#).

**Note:** Bits LAST, IPH and HOLD must not be asserted all at once

1. Missing a physical tooth naturally causes EAC to get into Halt mode.

### 35.15.7.2.2 TCR2 - Timer Counter 2

In Angle Mode, TCR2 (see the following figure) counts angle ticks instead of time.

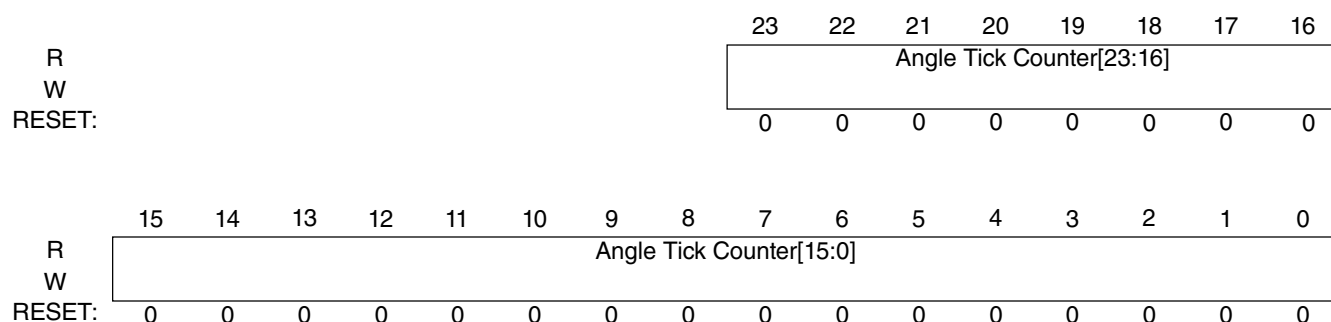


**Figure 35-321. TCR2 in Angle Mode**

This 24-bit free-running counter is used to generate an accumulated Angle Fraction value. It is updated by the Angle Tick Generator - refer to [Angle Tick Generator](#) for more details. Refer to [Count Control and High Rate Logic](#) for a detailed explanation about the use of this register in Angle Mode.

TCR2 provides continuous count of the angle in units of angle ticks. The Angle Tick Counter in TCR2 can be reset due to "Last Tooth" microcode indication. TCR2 Prescaling is disregarded in Angle Mode: physical tooth detection is done by EAC regardless of the value set in TCR2P.

### 35.15.7.2.3 TRR - Tick Rate Register

The exact period of the Angle Tick is programmed in the Tick Rate Register (see the figure below) by microcode. The TRR time unit depends on eTPU\_TBR\_TBCR\_ENG $n$ [ATGC] (see [Figure 35-324](#)):

- if ATGC=0, the TRR time unit is the period of TCR1 clocks taken as eTPU clocks divided by  $2 \cdot (\text{TCR1P} + 1)$ , even if TCR1CS=1 (see [eTPU Engine 1 Time Base Configuration Register \(eTPU\\_TBR\\_TBCR\\_ENG1\)](#) and [eTPU Engine 2 Time Base Configuration Register \(eTPU\\_TBR\\_TBCR\\_ENG2\)](#)).
- if ATGC=1, the TRR time unit is given by the eTPU clock divided by  $2 \cdot (\text{TCR2P} + 1)$ , using the TCR2 prescaler fed by eTPU clock/2 (independently of TCR1CS).

See [Calculating the Angle Tick Period Integer and Fraction](#) for a complete description about the mechanism to calculate the value to be written into this register.

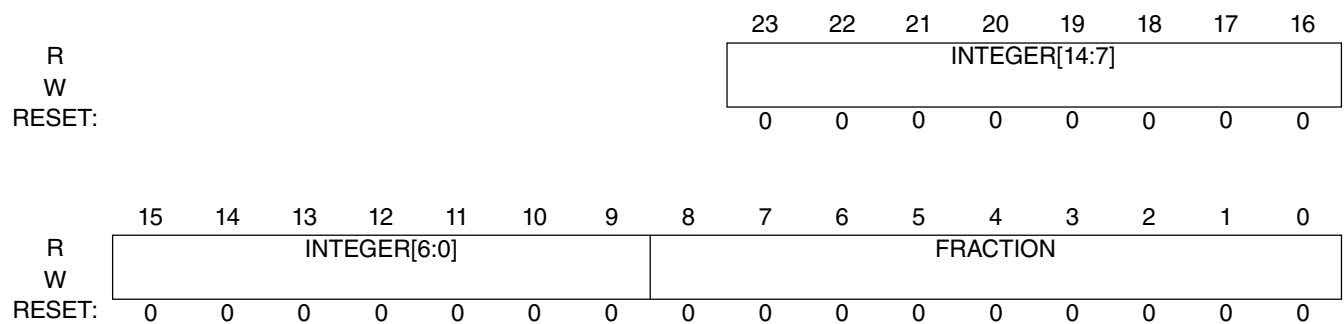


**Figure 35-322. TRR**

**INTEGER** — The integer part of TCR1/TCR2 clocks in one Angle Tick.

This number, decremented by one, works as a down-counter preload value. A value of **INTEGER=0** represents an integer of 32768. A new value written is reloaded into the counter (becoming effective) when a new tick starts or a tooth is detected or inserted via **IPH**.

**FRACTION** — Nine-bit fractional part of TCR1/TCR2 clocks in one angle tick.

The **FRACTION** value is accumulated in the **EAC Fraction Accumulator**, and whenever the result overflows (i.e., the accumulated fraction added up to an integer), the **Tick Prescaler** is halted for one **TRR** time unit.

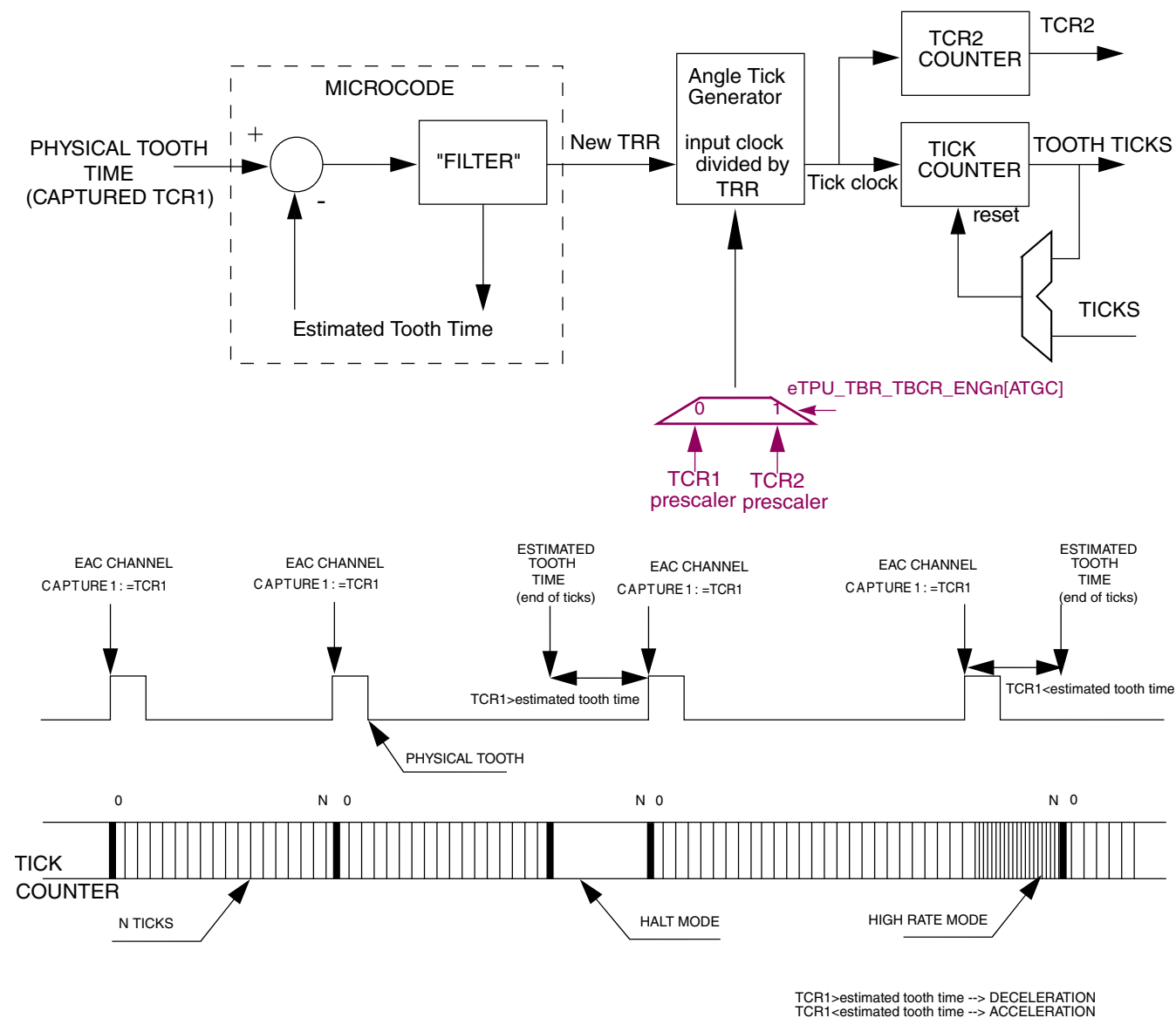


Figure 35-323. EAC "PLL"



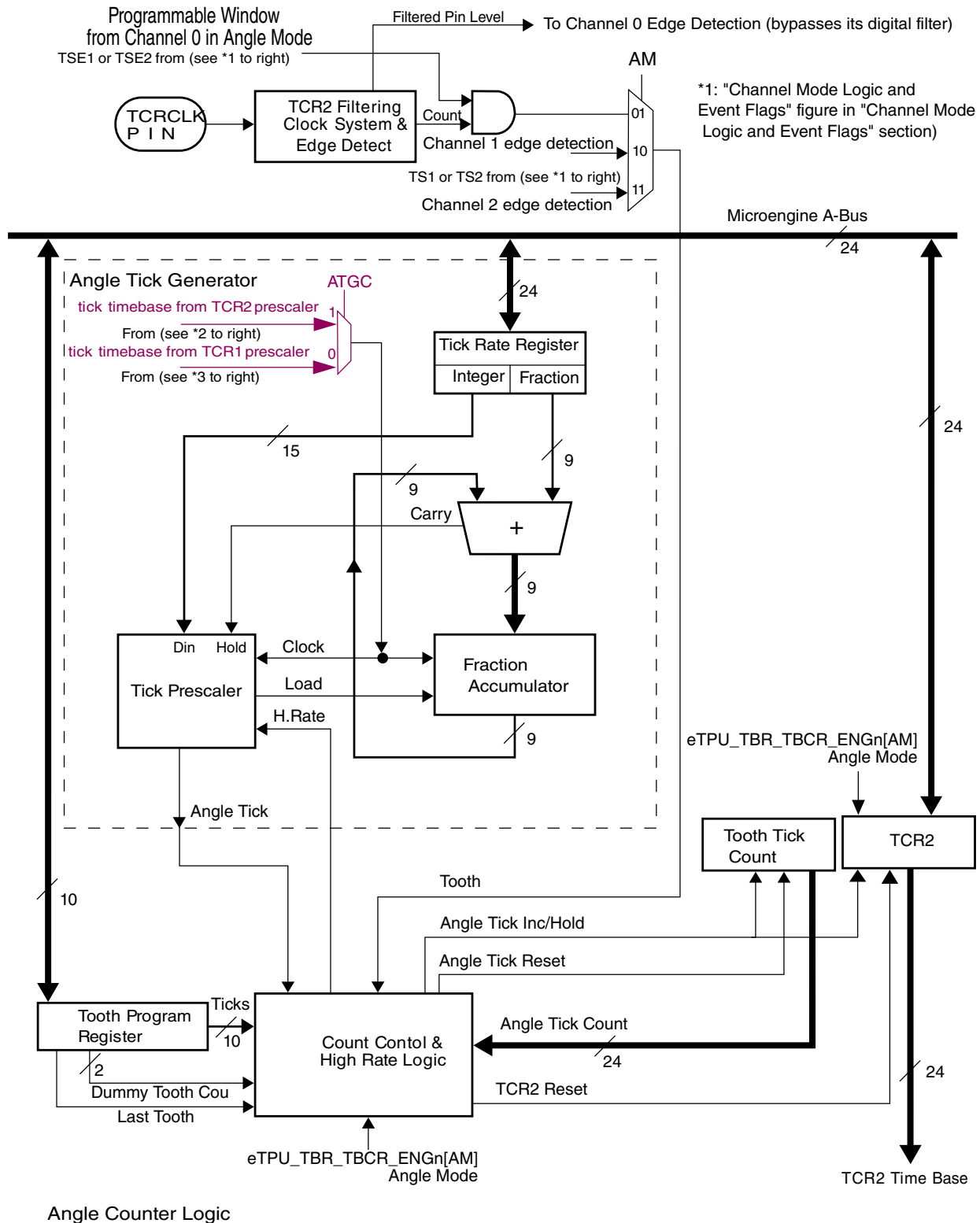


Figure 35-324. eTPU angle counter system

### 35.15.7.3 Acceleration and Deceleration

Acceleration and deceleration affect the new tooth period relative to the known period of the last tooth. Changes in tooth period may be extreme at very low engine RPM (such as cold start and warm start). The worst case of tooth period changes is caused during missing teeth, since there is more time for changes in angular velocity to be unnoticed by the EAC hardware. For example, on cold start (~20 RPM) there may be extreme acceleration: the ratio between a known tooth period before two missing teeth and the new tooth period after the missing teeth can be very high (up to a factor of 75). Acceleration and deceleration effects from tooth to tooth are less extreme as the engine climbs to high RPM.

In case of deceleration, the estimated tooth period ends before the actual tooth detection arrives. In this case, the EAC hardware waits at the end of the current tooth period, when it is said to be in **Halt mode**, until the real tooth indication is received, then continues with normal operation (**Normal mode**). See [Figure 35-323](#).

In case of acceleration, the actual tooth period is shorter than the estimated tooth period. As a result, a new physical tooth indication arrives before the end of the estimated tooth period. In this case the EAC closes the gap on **High Rate mode** by counting on eTPU clock divided by eight to the end of the tooth, advances to the next tooth, and switches back to normal operation mode. See [Figure 35-323](#).

The reason that the EAC does not jump directly to the next tooth is the need to provide sequential angle count throughout the whole tooth period, for channels or external STAC clients (if TCR2 is a STAC server) which compare angle in "equal" mode. These peripherals must get all the valid angle values in a sequential manner, to avoid missing angle matches.

TCR2 advancing from one tooth to another is a continuous count, and can be optionally reset at the end of the tooth. An estimated tooth is generated after the Tooth Tick Counter reaches the TICKS programmed value.

The EAC works continuously and switches automatically between Normal, Halt and High Rate modes. It relies on the microcode to calculate the estimated tooth period on every tooth, and to update the correct angle tick and tooth parameters in the EAC control registers. On high RPM, tooth period changes are reduced from tooth to tooth, and the EAC may follow the angle with good accuracy for several teeth without microcode intervention.

The EAC handles missing teeth by insertion of "dummy" teeth, or by enlarging the expected tooth period. It is a good practice to locate the flywheel missing teeth in non-critical angles, since a missing tooth may increase the angle measurement error (acceleration and deceleration is detected late).

### 35.15.7.4 Angle Tick Generator

The Angle Tick Generator is responsible for generating a programmed number of angle ticks in the tooth period. It generates the ticks in an average rate which ensures completion of the correct number of angle counts in the estimated period of the tooth, since the count of one tooth in angle ticks is independent on engine RPM. The main output of the Angle Tick Generator is the tick clock that feeds TCR2 in Angle Mode, as well as the internal Tooth Tick Counter (see [Figure 35-324](#)). The Tooth Tick Counter counts ticks within a tooth, from 0 up to TICKS, is controlled by the Angle Tick Generator logic and cannot be accessed by microcode. Refer to [Figure 35-325](#) for a generic presentation of the angle tick count and the measurement of a single tooth period.

#### 35.15.7.4.1 Calculating the Angle Tick Period Integer and Fraction

On each tooth the microcode has to update the exact period of a single angle tick used for counting the internal angle in the tooth. The period of an angle tick is determined by the EAC configuration, see [Generating the angle ticks](#) for details.

The microcode can use the eTPU MAC Divider unit (see [MAC and Divide Unit \(MDU\)](#)) to divide the tooth period by the number of angle ticks per tooth, which is stored in the TICKS field of TPR (refer to [Angle Mode registers](#)). This division yields the integer part of the angle tick period and the remainder. Dividing again the remainder shifted left nine positions, by the number of angle ticks per tooth translates the remainder to a 9-bit fraction. The microcode concatenates the 15-bit integer and the 9-bit fraction to a 24-bit value and writes it to TRR. The new rate is effective immediately after the next Angle Tick is generated by the Angle Tick Generator<sup>31</sup>.

For high RPM, note that shifting the tooth period value nine positions to the left prior to the first divide operation would calculate, in one operation, the integer and the fraction. For example: On 60 teeth flywheel running at 1000 RPM, tooth period is 1 msec. If TCR1/TCR2 counts @25 MHz, it counts 25,000 times in a tooth, which can be represented by 15 bits. Therefore the tooth period can be shifted nine positions to the left prior to divide operation, and be represented with 24 bits.

Using shift left nine positions and one divide operation would get the result in MACL register (in MDU) which holds the integer and nine bits of the fraction:

```
Angle_Tick_Rate {Integer[14:0], Fraction[8:0]} = (ToothPeriod <<9) / Ticks
TRR = Angle_Tick_Rate {Integer[14:0], Fraction[8:0]}
```

31. In High-rate mode, the tick keeps being updated at the rate of eTPU clock/8 until it goes back to Normal mode, when the new TRR value is used.

**NOTE**

In the pseudocode above, the tooth period (ToothPeriod) is not, in general, the value of estimated tooth time. It is obtained by microcode by subtracting TCR1 values between two teeth detections, converted to TCR2 periods if ATGC=1. Its comparison with the estimated tooth time indicates acceleration (if minor) or deceleration (if greater) to the microcode.

On low RPM the initial tooth period, measured in TCR1/TCR2 counts, may be too big to be shifted nine positions to the left. For lower RPM (for example 500 RPM) the tooth period cannot be represented by 15 bits, and shifting it nine positions to the left would lose the MSB. In this case, two divide operations are required as follows: first divide the Tooth Period by the number of TICKS - the integer is stored in MACL and remainder in MACH. MACL is saved in another register. MACH is shifted 9 positions to the left and divided again by TICKS. In parallel with the second divide, the register which saved the original MACL is shifted left 9 positions. After the divide MACL contains the 9 bits fraction and the other register contains the 15-bit integer, shifted left nine times. The logical OR of the two registers is written to the TRR:

```
Angle_Tick_Rate {Integer[14:0]} = (ToothPeriod) / Ticks
Remainder[9:0] = (ToothPeriod) mod Ticks
Angle_Tick_Rate {Fraction[8:0]} = (Remainder[9:0] << 9) / Ticks
TRR = Angle_Tick_Rate {Integer[14:0], Fraction[8:0]} = (Integer[14:0] << 9) | Fraction[8:0]
```

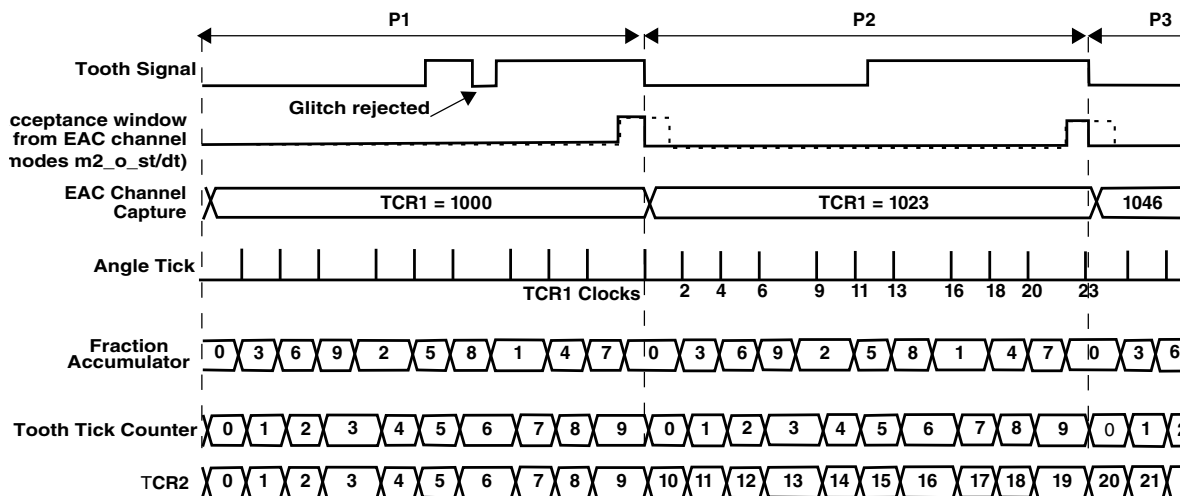


**Figure 35-325. Angle Ticks Generation**

### 35.15.7.4.2 Generating the angle ticks

The integer part of TRR is preloaded to a prescaler, which counts in time intervals as follows (see [Figure 35-324](#)):

- If  $\text{eTPU\_TBR\_TBCR\_ENGn[ATGC]}=0$ , one unit of TCR1 clock period (if  $\text{TCR1CS}=0$ ) or twice the TCR1 clock period (if  $\text{TCR1CS}=1$ )<sup>32</sup>, that is,  $\text{eTPU clock period} \times 2 \times (\text{TCR1P}+1)$  (independently of TCR1CS).
- If  $\text{ATGC}=1$ , two units of the eTPU clock period multiplied by the TCR2 prescaler factor, that is,  $\text{eTPU clock period} \times 2 \times (\text{TCR2P}+1)$ .

When the down counter reaches zero, it generates an angle tick pulse to the Angle Counter Logic and a Load pulse to the Fraction Accumulator. It is then preloaded with most updated TRR integer part. Due to the Load pulse, the 9-bit fraction is accumulated in a 9-bit Fraction Accumulator. If a fraction overflow condition occurs (the 9-bit adder asserts carry out), the accumulator saves the lower 9 bits of the addition result, which is the remaining fractional part. The carry out bit indicates an accumulated integer "one" which means that the angle tick is early by one input clock. It halts the prescaler operation for one input clock to compensate the accumulated error generated by the integer prescaler. As a result, the average angle tick period takes into account both the integer and the fraction parts. The accuracy depends on the bit count of the fraction. Using 9-bit fraction part while the width of the field TICKS in register TPR is 10 bits provides accuracy of two LSB on a full scale ( $\text{TICKS}=1023$ ) or one LSB on lower scale ( $\text{TICKS} \leq 511$ ).

When the Tick Prescaler gets High Rate mode indication from the Angle Counter Logic, it generates angle ticks at a rate of eTPU clock divided by eight. In this case it does not generate Load pulses to the Fraction Accumulator, ignores its "hold" input and preloads internally to a fixed period of eight eTPU clocks. When High Rate mode is entered, the prescaler is preloaded to a period of eight eTPU clocks before its first angle tick generation, ensuring separation of at least eight eTPU clocks between the last Normal mode angle tick and the first High Rate mode angle tick. The fraction accumulator resets when the tick count advances to the next tooth, or when TRR is written by the microcode.

### 35.15.7.5 Count Control and High Rate Logic

The Count Control and High Rate Logic controls TCR2 operation in Angle Mode, using the angle ticks generated by the Angle Tick Generator. Count Control logic is responsible for advancing, holding and resetting the TCR2 and Tooth Tick Counter in the proper timing, such that the TCR2 time base will reflect the correct estimated angle. This logic also includes the Tooth Program Register (TPR, see [Angle Mode registers](#) for more information).

---

32. The EAC logic overall runs at a rate of eTPU clock divided by 2, therefore the tick generator takes the halved TCR1 clock when  $\text{TCR1CS}=1$  and  $\text{ATGC}=0$ .

The Count Control and High Rate Logic handles deceleration, acceleration, missing teeth and last tooth. On High Rate (acceleration) it ensures that the angle bus scans all valid angle values in a rate which can be traced by the STAC bus. This operation enables external STAC clients (if TCR2 is a STAC server) or channels working in "equal-only" comparator mode to match the TCR2 exported angle information in "equal" mode, in an exact match.

Because the eTPU channels are capable of capturing either TCR1 or TCR2 due to signal transition, the microcode can get either the angle or time of the physical pin transition. Since the EAC channel (0, 1 or 2) is connected to the physical tooth, the microcode can get the EAC error in angle domain (tooth appears at the wrong angle) or time domain (physical tooth captured time into the EAC channel, relative to the estimated tooth time). Note that in angle mode, the transition detect logic of the channel 0, if selected as the EAC channel, is fed from the digital filter of the TCRCLK signal, and not from the channel 0 internal digital filter. This ensures synchronous operation of channel 0 and the EAC hardware.

Another feature of the eTPU channel, when working in Single Match And Single Transition Enhanced Mode (refer to [Single Match Enhanced Mode \(sm\\_st\\_e\)](#)), is capturing a single time base due to signal transition before and after the digital filter. This option allows subtracting the digital filter delay to get accurate signal transition timing on the channel. This way, the TCRCLK signal may be programmed with a slow and reliable digital filter, and get accurate time measurement of the digital filter delay.

To assert the end of the estimated tooth period the Count Control and High Rate logic compares the TICKS field in TPR (refer to [Angle Mode registers](#)) with the current value of the tooth tick counter. When the Tooth Tick value is greater or equal to TICKS, it determines the end of the estimated tooth period. On acceleration this event occurs during High Rate mode operation, after the arrival of a physical tooth. In deceleration, this event occurs during Normal Mode, before the arrival of a physical tooth. On constant angular velocity, this event appears together with the arrival of a physical tooth.

The following sections describe the operation of the Counter Control and High Rate logic.

#### 35.15.7.5.1 Normal Mode

In Normal mode the Counter Control logic advances TCR2 and the Tooth Tick counter as if the engine has a constant speed during the tooth period. It receives the angle ticks from the Angle Tick Generator in an average rate which is determined by the Tooth Rate Register (TRR). This is the reset mode.

When the Tooth Tick Counter is about to reach the last value effectively stored in TPR field TICKS plus one, the hardware detects the end of the estimated tooth period. If the physical tooth and the estimated tooth arrive at the same time the EAC stays in Normal mode, the Tooth Tick Counter is reset, and TCR2 is incremented (depending on TPR bits LAST and MISSCNT). If the physical tooth and the estimated tooth do not arrive at the same time, either acceleration or deceleration is detected, and the EAC switches to the proper mode. See the figure below for a detailed diagram of Normal Mode behavior.

The microcode which services the EAC channel physical tooth transition may update TRR according to various conditions to give the best estimation of the current tooth period, according to the previous tooth period and other engine parameters.

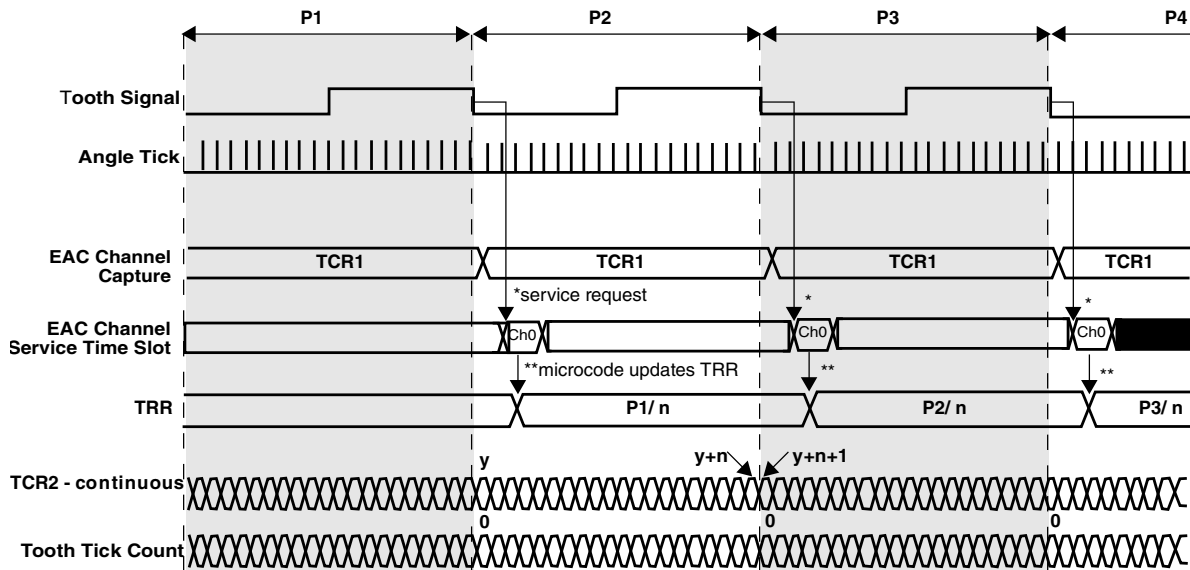


Figure 35-326. Normal Mode

### 35.15.7.5.2 Halt Mode (Deceleration)

In case of deceleration, the Tooth Tick Counter reaches the TICKS value before the arrival of the next tooth. The Count Control logic does not reset neither advances TCR2 and Tooth Tick Counters. The Count Control logic halts TCR2 and Tooth Tick Counters at the end of the tooth, waiting for the physical tooth to arrive.

When the physical tooth is detected the EAC switches back to Normal Mode and releases TCR2 to count the angle ticks of the new tooth, also resetting the Tooth Tick Counter. Only then TCR2 may wrap to 0, if TPR bit LAST is asserted. See the figure below for a detailed diagram of Halt Mode behavior.

The microcode service caused by the physical tooth determines the deceleration, calculates the new tooth period and Angle Tick period and updates TRR. This operation slows the angle tick rate generated by the Angle Tick Generator on-the-fly, to the rate required for the new tooth period.

Since the microcode service is initiated by the physical tooth edge, microcode latency may introduce a small angle error caused by using the TRR value of the previous tooth at the beginning of the current tooth. On high RPM, deceleration is relatively small but the microcode latency may take a significant percentage of the tooth period. On low RPM microcode service latency takes little percentage of the tooth period, but there may be cases of extreme acceleration and deceleration. The microcode latency can be calculated knowing TCR1 value during the service time, and TCR1 value captured in the EAC channel due to the physical tooth pin transition. The duration of the Halt mode is obtained using the estimated tooth time.

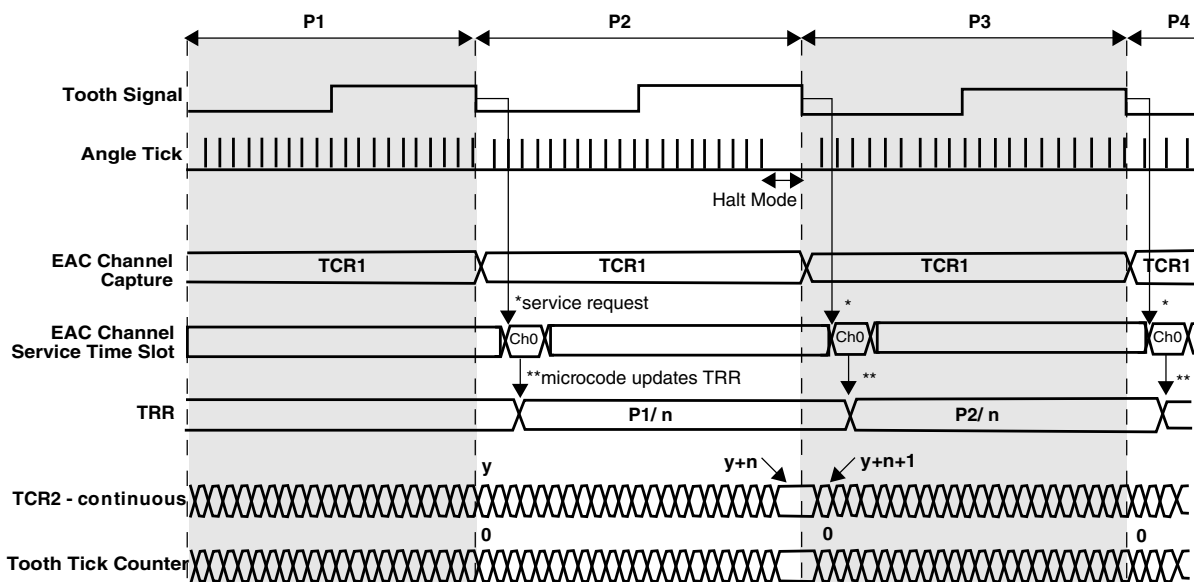


Figure 35-327. Halt Mode - Deceleration

### 35.15.7.5.3 High Rate Mode (Acceleration)

In case of acceleration, the next tooth arrives before the Tooth Tick Counter reaches the TICKS value. In this case the High Rate logic is responsible for closing the gap and advancing the tooth on the correct timing. The High Rate mode operates as follows:

- When the acceleration is detected (physical tooth arrives before the Tooth Tick Counter reaches the TICKS value), the Count Control and High Rate logic switches to High Rate mode in which both the Tooth Tick Counter and TCR2 count at rate of



eTPU clock divided by eight, until the Tooth Tick Counter reaches the current TICKS value. To ensure correct operation, the TICKS value is sampled in the logic at the beginning of the mode.

- At this point, which represents the estimated tooth edge, the logic resets the Tooth Tick Counter and advances TCR2 (or resets it if LAST is asserted and MISSCNT = 0).
- The control logic switches back to Normal Mode, using the most updated TRR value as input to the Angle Tick Generator. The logic samples the updated TICKS value for the tooth estimation, last tooth indication and number of missing teeth from TPR.

In High Rate mode the angle ticks are provided at high speed until the end of the current tooth. This operation is required to scan all the valid angle values of the current tooth, in a rate which is not too high for the STAC bus continuous update, but much higher than the rate dictated by TRR.

EAC channel microcode, which services the physical tooth transition detection, can start its service either before High Rate mode operation is complete (the Tooth Tick Counter has not reached the TICKS value) or after the EAC switched back to Normal mode. Any physical teeth received while the EAC is in High Rate Mode does not alter the immediate EAC state, but it is still detected by the EAC channel logic and can, therefore, alter future EAC behavior (for instance, closing the tooth detection window - see [Angle Logic and Channel Modes](#)).

At the beginning of High Rate mode operation, the TPR value is preloaded into a temporary register in the Counter Control Logic, used for scanning all the valid values to the end of the current tooth, with its appropriate LAST and MISSCNT attributes. While the EAC is in High Rate mode operation, the effect of microcode update of TPR fields LAST, MISSCNT and TICKS is delayed to the next estimated tooth, after the High Rate mode operation is complete (see [Special TPR write cases](#)). This is because the current physical tooth represents the next estimated tooth. If the microcode updates this field after High Rate mode operation is complete, the current physical tooth and estimated tooth are the same, and the effect is immediate. Either in High-Rate mode or not, the value read by microengine is the same written, even if not yet effective, until the EAC resets LAST and/or IPH, or decrements MISSCNT. Typically the microcode service may occur during the High Rate mode on extreme acceleration situation at low RPM. Therefore, the microcode operations are always related to the real physical tooth. From the above it can be seen that the microcode updates of the TICKS field in TPR affect the end time of the current physical tooth. For correct operation, this field should be updated before the Tooth Tick Counter has reached either the old or the new TICKS value.

During High Rate mode operation, TRR is ignored and the Angle Tick Generator uses eTPU clock divided by eight. Therefore, the TRR update by microcode will take effect only after the EAC switches back to Normal mode. If microcode service occurs after the Tooth Tick Counter has been reset, the EAC is already back in Normal mode, and some angle ticks may have been counted at the rate of the previous tooth. In this case the new TRR value will have immediate effect on the angle tick period, and the microcode should take into consideration the delay from the physical tooth to the estimated tooth in calculation of the next tooth period. See the figure below for a detailed diagram of High Rate Mode behavior.

An angle error may be introduced by the duration of the High Rate mode. Also, the scheduler latency may introduce a small accumulated error by using TRR value of the previous estimated tooth at the beginning of the current tooth. After the estimated tooth has advanced, the duration of the High Rate mode operation is the actual delay from the physical tooth edge to the estimated tooth edge. This delay can be obtained by comparing the estimated tooth time with the EAC channel capture register which captured TCR1 on the physical pin transition.

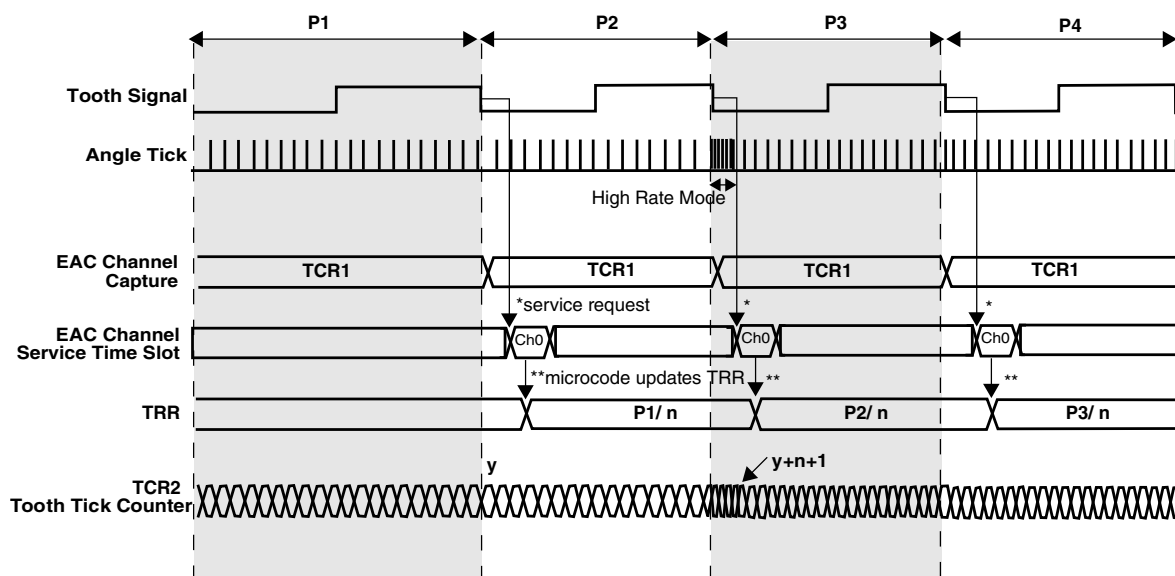


Figure 35-328. High Rate Mode - Acceleration

### 35.15.7.6 Special Cases of Missing Teeth and Last Tooth

The EAC handles cases of up to three missing teeth and the last tooth in the engine cycle. The following paragraphs describe these functions.

### 35.15.7.6.1 Handling the Last Tooth

The microcode can set the TCR2 counter to work in engine periods (wrap-around count) or continuous angle measurement.

For periodic operation, during the last engine cycle tooth the EAC microcode has to set the LAST flag in TPR. As a result, when the tooth period is ended, the Counter Control Logic generates a reset command to both the Tooth Tick Counter and TCR2 instead of an advance command. The operation resets the TCR2 based angle count, indicating a new period of the engine cycle. This implementation provides an engine cycle based periodic angle measurement.

### 35.15.7.6.2 Handling Missing Teeth

The EAC can handle up to three missing teeth in two ways:

- Count the angle ticks relative to the last physical tooth. The microcode should update the TPR TICKS field to the number of angle ticks included in two, three or four teeth, according to the flywheel type (one, two or three missing teeth). EAC hardware works in its regular manner.
- Insert a "dummy" tooth instead of the missing tooth, at the estimated point in time. After the "dummy" tooth, the Angle Tick Counter is incremented as if there was a physical tooth. A "dummy" tooth can be inserted only during Normal or High Rate operation modes. The microcode inserts "dummy" teeth by writing to the MISSCNT field in TPR.

In the first option the missing tooth is not counted on the angle measurement. For example, a flywheel with 59 physical teeth and one missing tooth can be considered as 58 identical teeth numbered (0-57) and tooth number 58 has a double number of angle TICKS. In this case a 720 degrees engine cycle has 118 teeth. TCR2 reflects the real angle, since it counts angle ticks continuously.

In the second option, the missing teeth are counted as "regular" teeth by automatic insertion of "dummy" teeth. The microcode has to write a non-zero value to the MISSCNT field in TPR. This field is a 3-bit down counter which affects the operation of the Counter Control logic.

For example, a flywheel with 59 physical teeth (0-58) and one missing tooth (59) can be considered as 60 teeth numbered (0-59), all having the same number of angle ticks. The microcode has to write "001" to the MISSCNT bits during the period of tooth number 58 to indicate that next tooth (59) is missing.

When the Tooth Tick Counter reaches the TICKS value, TCR2 is incremented as if a physical tooth has been detected. In addition, the MISSCNT value initializes a "dummy tooth counter" which is decremented to indicate the number of left "dummy teeth" which still need to be generated. Because a dummy tooth was counted, EAC does not enter Halt Mode and Tooth Tick Counter continues incrementing in the absence of a physical tooth detection.

In case of extreme acceleration on very low RPM (cold start) there can be a situation that the first physical tooth after one or two missing teeth appears even before the "dummy" tooth is generated. Due to the acceleration the EAC switches to High Rate mode in order to run through all the valid angle values, including the dummy teeth. When the Tooth Tick Counter reaches the TICKS value on High Rate mode, and the "dummy tooth" down counter is not zero, the generated "dummy tooth" advances to the next tooth and decrements the "dummy tooth" counter, but does not switch the EAC back to Normal mode. The last "dummy tooth" decrements the counter to zero, indicating that no more dummy teeth are to be inserted, and the next tooth is an estimated physical tooth. The EAC continues at High Rate mode until the Tooth Tick Counter reaches the TICKS value again, then advances to the next tooth while switching back to Normal mode. When in High Rate mode, the TPR register does not reflect the MISSCNT downcounting, see [TPR Buffering](#) for details.

MISSCNT can be rewritten before it reaches 0, allowing it to count more than 7 missing teeth, as long as no physical tooth arrives between the first MISSCNT write and the rewrite.

### 35.15.7.6.3 Combining Missing Teeth and Last Tooth

The Last Tooth indication takes effect when there are no more missing teeth to be generated, i.e the "dummy tooth" counter value is zero. If, for example, the microcode sets the missing teeth counter to "10" (two missing teeth) and sets the LAST flag, the first and the second dummy teeth will increment TCR2, and the third estimated tooth, which correlates with the physical tooth (the first of the next cycle), will reset TCR2, because LAST was set. This scheme enables the microcode to define one or more missing teeth to be replaced by "dummy tooth" insertion, and the end of the engine cycle in one service request. It is assumed that the two missing teeth must come together in the same engine cycle, and not split between two engine cycles (either the missing teeth are both last in an engine cycle or both not last, but not last in one engine cycle and first in the next).

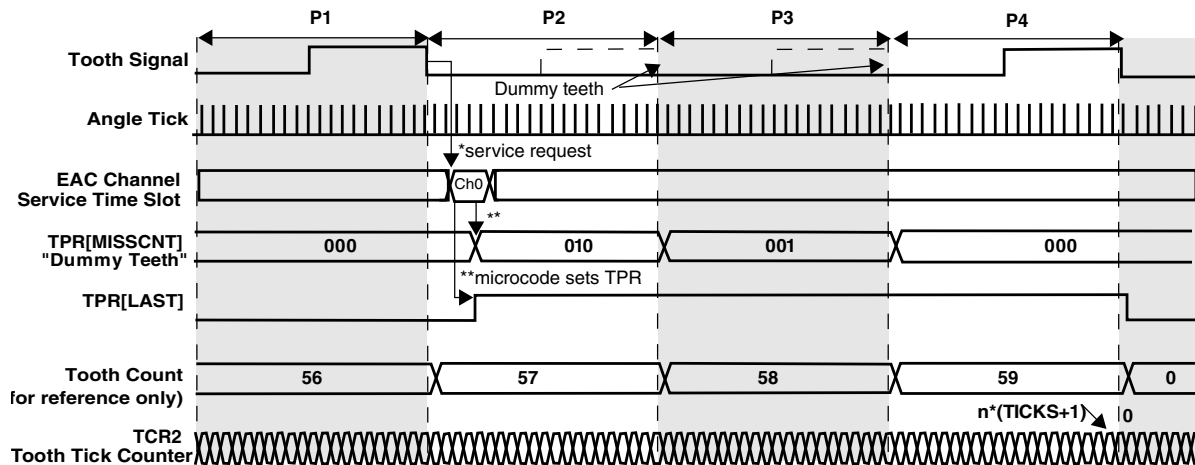


Figure 35-329. Missing Teeth and Last Tooth Combination

### 35.15.7.7 Handling Mechanical Tooth Correction

The EAC can handle tooth edge detection errors caused by flywheel mechanical errors. The eTPU application can hold a vector of tooth mechanical errors with one entry per tooth. This error can be measured in angle ticks which are independent of engine RPM. The TRR can be updated to the fixed period of any tooth, including its mechanical error.

Because TCR2 counts continuously, without being reset, the mechanical correction is transparent. Though the tooth has its own programmed TICKS value, TCR2 simply counts angle ticks, disregarding the boundary between two adjacent teeth.

### 35.15.7.8 Handling Mis-detected Tooth

When a physical tooth signal is missed by the engine sensor, the EAC may get into Halt mode at the end of the estimated tooth period, expecting the physical arrival. In this case, a Match timeout event of EAC channel will call service which detects extreme deceleration. The microcode can assert the IPH bit in TPR, to force the detection of the missed physical tooth. It can also calculate the accumulated angle bus error, and fix the next estimated tooth period, to close the gap.

### 35.15.7.9 Handling False Tooth Detection

Most of the false tooth detection, caused by noises on the engine tooth sensor, can be eliminated by the window blanking filtering, timed by EAC channel match recognitions. The EAC also provides means of fixing false detection of an additional tooth which passed the window filter. When such an event occurs, the EAC switched to High Rate

mode (advancing to the next tooth) and when the next physical tooth arrives, an extreme acceleration is detected: the EAC sees the remaining portion of the current tooth period as another tooth period. The microcode can detect the situation when the acceleration is not realistic, or when immediately after the detection of this extreme acceleration, the following tooth indicates extreme deceleration back to the original RPM.

When the microcode detects such a case, the Tooth Counter has been advanced by mistake to the next tooth. The microcode can set the HOLD bit in the TPR, forcing the EAC to freeze and wait for the next physical tooth to close the gap. When the next physical tooth arrives, HOLD is automatically negated and the EAC proceeds from that point to the remaining portion of the tooth period, in the same mode it was when HOLD bit was asserted.

### 35.15.7.10 Angle Logic and Channel Modes

The TCRCLK transition detection is qualified by a signal that comes from channel 0 (see [Figure 35-324](#)) and depends on the particular mode (PDCM) programmed for that channel. This configures a window for TCRCLK detection for the angle logic which is the same (except on High Rate mode, see [High Rate Mode \(Acceleration\)](#)) used to set the TDLA flag on single transition modes, and TDLB on double transition modes (see signals TSE1, TSE2 in [Figure 35-307](#)). The same applies when channels 1 or 2 are used to control EAC (see signals TS1, TS2 in [Figure 35-307](#)). As a consequence, the window depends on the channel mode as follows:

- on all modes, the window closes upon a tooth edge detection: TDLA asserted on single transition modes, TDLB asserted on double transition modes.
- on mode m2\_st: the window opens on Match A (which enables Transition A) and does not close with Match B. If Match B comes before Match A, it blocks Match A and, hence, Transition A.
- on mode m2\_o\_st: the window opens on Match A (which enables Transition A) and closes on Match B. Match B is enabled by Match A, so it cannot come before.
- on all other Single Transition modes, the window is "always open", independently of matches.
- on mode m2\_dt: the window opens on Transition A, which is enabled by Match A. The window does not close with Match B, but if it comes before Match A the later gets blocked and, hence, blocks Transitions. Match A is also a condition for the window, so the microcode closes it by clearing MRLA.

- on mode m2\_o\_dt: the window opens on Transition A, which is enabled by Match A. The window closes on Match B, which is enabled by Match A. Match A is also a condition for the window, so the microcode closes it by clearing MRLA.
- on all other Double Transition modes, the window opens on Transition A.

### 35.15.7.11 Restarting angle logic

It is not recommended to change the eTPU\_TBR\_TBCR\_ENGn[AM] field while GTBE=1. However, if the angle logic must be restarted without interfering with the timebase count running on TCR1, the procedure below must be followed:

1. Program eTPU\_TBR\_TBCR\_ENGn[AM]=00 and TCR2CTL=111 at once. That prevents TCR2 from incrementing while the angle logic is disabled. The angle logic state machine resets to Normal mode and the tick prescaler to the initial count with AM=00, but not the microengine registers TPR and TRR.
2. Start a thread to reconfigure the EAC. The thread must set the EAC controlling channel (0, 1 or 2) flags in a state, depending on the channel mode, that lets the channel tooth detection window open (see [Angle Logic and Channel Modes](#)). It can optionally write TCR2 with an angle preset value equivalent to the first tooth expected after restart. The thread must also set TPR bit HOLD=1. The TPR bit IPH must be 0.
3. After the thread is finished, program eTPU\_TBR\_TBCR\_ENGn[AM]=01, 10 or 11, and TCR2CTL according to the desired tooth edge selection if AM=01.

The first tooth detected after this procedure restarts the TCR2 counting, unfreezing the Angle Mode logic into normal mode.

### 35.15.7.12 Special TPR write cases

This section describes how simultaneous modification of TPR fields are resolved, and how the effect of TPR writes depend on the EAC mode.

#### 35.15.7.12.1 TPR Buffering

In High Rate mode (see [High Rate Mode \(Acceleration\)](#)), TPR writes are immediately effective only for bits IPH and HOLD. Writes to all other fields are "buffered" and become effective when EAC leaves High Rate mode. However, if TPR is written a

second time right after IPH is asserted in Normal mode, this second write behaves as if EAC is still in Normal mode. Only in the next microcycle (after execution of a nop, for instance) the TPR writes are buffered, acknowledging High Rate mode.

MISSCNT and LAST can be written any value during High Rate mode, and the value that prevails for the next tooth is the one sampled when EAC goes back to Normal mode (or the value written in Normal or Halt mode thereafter). If MISSCNT and/or LAST are not zero when High Rate mode begins, they are sampled into the internal EAC logic and are effective while High Rate lasts (missing teeth count continues, and TCR2 is reset at the end of High Rate if LAST=1). However, their values in TPR are reset when High Rate mode starts. After that and until the end of High Rate mode, the value read by microcode is the same written. This behavior prevents read-modify-writes to TPR from unwillingly rewriting LAST or MISSCNT.

### **35.15.7.12.2 IPH and LAST**

If both IPH and LAST are asserted in the same microinstruction with HOLD=0, the EAC acts as if LAST was set first and then IPH right after, so that:

- in Normal mode, it goes to High Rate with LAST=1;
- in Halt mode, it goes to Normal Mode resetting LAST (and TCR2);
- in High Rate mode, IPH is ignored and LAST becomes effective in the next tooth (physical or inserted) after it goes back to Normal mode.

### **35.15.7.12.3 IPH and TICKS**

Because of different results depending on the EAC mode at the time of TPR write, it is not advisable to write 1 to IPH and change TICKS at the same microinstruction. A consistent behavior is obtained if IPH is written first and TICKS on the second microinstruction after (for instance with a nop between them), making the new TICKS value valid for the next tooth regardless of the mode.

The mode-dependent behavior is (assuming HOLD=0):

- in Normal mode, the new TICKS value becomes valid before EAC goes High Rate due to the IPH;
- in Halt mode, the EAC goes to Normal mode, and new TICKS is valid for the next tooth;
- in High Rate mode, the new TICKS value is effective when EAC leaves High Rate mode, and IPH is ignored;



### 35.15.7.12.4 IPH and MISSCNT

If both IPH and MISSCNT are written non-zero values with HOLD=0:

- in Normal mode, at the next microcycle the EAC goes to High Rate mode, the MISSCNT field in TPR goes to 0, and the missing teeth are counted in High Rate mode.
- in Halt mode, the EAC goes to Normal mode for one microcycle and then, yet another microcycle later, goes to High Rate mode, counting the missing teeth. The TPR fields IPH and MISSCNT are zeroed on the transition from Normal to High Rate mode.
- in High Rate mode, IPH is ignored (resetting at the next microcycle) and MISSCNT is buffered (see [TPR Buffering](#)).

### 35.15.7.12.5 IPH and HOLD

If IPH and HOLD are asserted at once, IPH cancels the HOLD and both reset. The EAC is not frozen, regardless of the EAC mode (High Rate, Normal or Halt).

### 35.15.7.12.6 LAST and HOLD

If LAST and HOLD are written 1 at once, LAST asserts and EAC is frozen. Afterwards, when a physical tooth is detected or IPH is asserted, the EAC is unfrozen in the same state it was before, and LAST is kept asserted.

## 35.15.8 Microengine

Each eTPU engine has a microengine that fetches, decodes and executes microinstructions. The Microengine only works when there are service requests to be attended, otherwise it turns to idle state, controlled by Hardware Scheduler (see [Scheduler](#)).

Microcode is stored in Shared Code Memory (SCM) that is 32-bit wide. Microengine can access SCM using a different bus from the one used to access code memory, so that code and data can be accessed at the same time (Harvard Architecture).

Some of eTPU functionality can only be made through the microengine, like configuring channels and interrupting host. Microengine gives eTPU a high degree of flexibility, since any desired treatment for channel's events can be implemented; however, that

flexibility comes at the cost of channel service's latency. Latency is worsened when channels from a same eTPU engine contend for microengine service. In the figure below, a block diagram of microengine architecture is shown.

Follows a summary of Microengine features:

- P, DIOB, A, B, C, D, SR, RAR, LINK, CHAN, MACL, MACH, ERTA, ERTB, TCR1, TCR2, TPR, TRR registers are accessible by microcode.
- 24-bit ALU and Post-ALU shifter performs basic arithmetic and logical operations described in [ALU and Post-ALU Shifter](#).
- MDU (MAC/Divide Unit) performs integer MAC, multiply and divide operations.
- Fixed Microinstruction Size of 32bits.
- Fixed-length instruction execution (2 eTPU clocks)
- Static superscalar operation

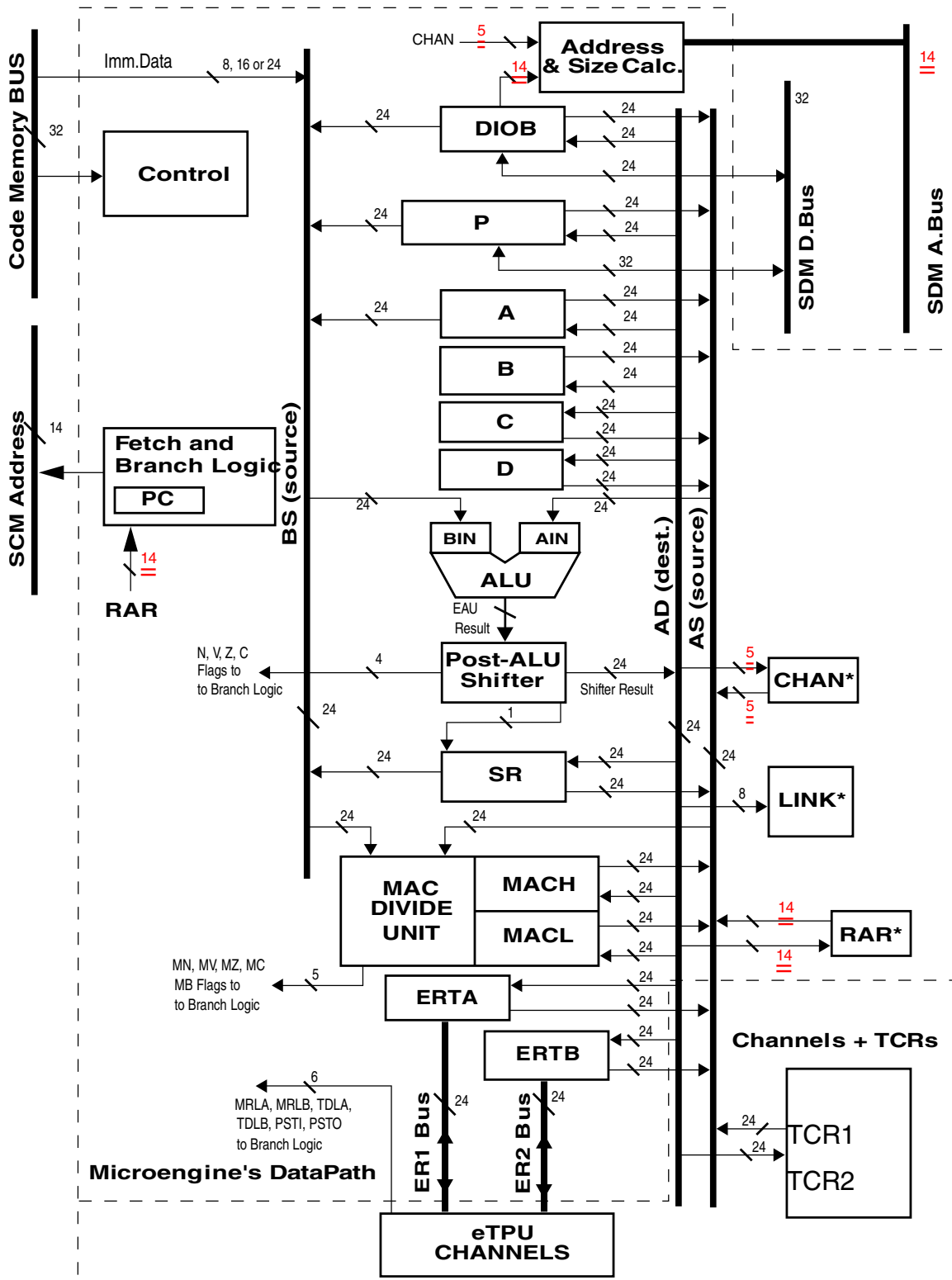


Figure 35-330. Microengine Block Diagram

### 35.15.8.1 Registers

eTPU microengine accesses a total of 18 registers. Fourteen of them are special purpose (registers A, B, C and D are for general use). Special purpose registers except CHAN and LINK can also be used as general use if the operation that use their contents are not performed. Register description is intended to just introduce their functionalities and not to provide detailed explanation of it since it will be described in [Microinstruction Set](#). Registers less than 24 bits in size are right-justified.

None of the registers have guaranteed reset values. However, some are initialized just before the thread starts (see [Time Slot Transition](#)).

#### 35.15.8.1.1 P Register

P register is the only one that is 32-bit wide. It can be used as source and destination for arithmetic/logical operation, and as source and destination for SDM read/write operations.

For P source/destination possibilities in ALU/MDU microoperations, see [Selecting sources and destination](#).

When P is used as SDM read/write operations source or destination there are only 3 possibilities of access: all 32 bits, lower 24 bits and upper 8 bits. SDM operations are explained in detail in [SDM Microoperations](#).

P is automatically loaded with one parameter before the thread starts (parameter preload). For more information see [Entry Point format](#) and [Time Slot Transition](#).

Upper 8 bits of P register can be used as application state, since these bits can be tested as branch conditions. P[31:24] is also used in dispatch microoperation (see [Dispatch Microoperation](#)), and bit pairs P[29:28], P[27:26], P[25:24] can be directly copied into Channel flags 1 and 0 using field FLC. Together with Entry Table Condition Encoding, it provides fast state resolution without code execution.

#### 35.15.8.1.2 DIOB Register

DIOB is an abbreviation for Data Input/Output Buffer. DIOB is 24-bit wide and can be used as source and destination for arithmetic/logical operations as well as SDM data source and destination. DIOB only can be accessed as 24 bits, both in arithmetic/logical and SDM read/write operations. When using DIOB to perform an SDM access, only lower 24 bits of SDM will be accessible (SDM upper 8 bits always remains unchanged).

DIOB can also be used as SDM addressing register, when DIOB contents is used as absolute SDM address (14-bit wide). In this case DIOB can also be pre-decremented or post-incremented (see [Indirect Addressing Mode](#)).

DIOB is automatically loaded with one parameter before the thread starts (parameter preload). For more information see [Entry Point format](#) and [Time Slot Transition](#).

### 35.15.8.1.3 ERTA and ERTB Registers

ERTA/B registers are 24-bit wide and can be used as source or destination in arithmetic/logical operations. ERTA/B are the only source for channel's match registers write (see [Write Channel Match and UDCM Registers](#)). ERTA can also be the source for UDCM write.

When a thread starts to be executed, ERTA and ERTB are loaded with a copy of CaptureA and CaptureB registers respectively. ERTA/B can be used to receive a copy of MatchA and MatchB registers. ERTA/B are the only destination of MatchA/B read operation (see [Special T4ABS Source Operation: Read Match Registers](#)).

ERTA and ERTB also receive a copy of CaptureA and CaptureB registers when CHAN register is written (see [CHAN Register](#)). For more information about Capture and Match registers see [MatchA and MatchB Registers](#) and [CaptureA and CaptureB Registers](#).

### 35.15.8.1.4 SR register - Shift Register

SR is a 24-bit wide register that can be used as source and destination register for arithmetic/logical operations. SR can shift right its contents by 1 bit at time and, at the same time, receive in its bit 23 the lost bit of a shift-right operation in post-ALU shifter ([ALU and Post-ALU Shifter](#)), allowing SR to be used to perform 48-bit shift right (see [Shift Operations](#)).

### 35.15.8.1.5 MACH and MACL Registers

Both MACH and MACL are 24-bit registers, part of MAC/Divide unit (see [MAC and Divide Unit \(MDU\)](#)). They can be used as source and destination in most arithmetic/logic operations. When multiply or divide operations are used (multiply-accumulate included), MACH and MACL have special purpose and some restrictions apply, see [MAC and Divide Unit \(MDU\)](#) for more information.

### 35.15.8.1.6 LINK Register

Link Register is an 8-bit wide register and can be used only as destination in arithmetic operations. LINK is a write-only command register, which precludes its use as a source register for ALU operations. When LINK register is written, it issues a service request for the channel number and eTPU engine equal to the number written in LINK register (see [Functions and threads](#) and [Channel Link](#) for information about Link Service Request).

### 35.15.8.1.7 RAR Register

RAR is a 14-bit register and can be used as source and destination in arithmetic operations. RAR also receives the contents of PC register when a subroutine call is executed. The contents of RAR register is loaded into PC when a return from subroutine is executed. RAR is loaded with value 0x3FFF during TST. For more information about subroutine call and return see [Branch Operations](#) and [Return from subroutine](#) respectively.

### 35.15.8.1.8 CHAN Register

CHAN is a 5-bit register that can be used as source and destination in arithmetic operations. The contents of CHAN register affects the execution of many channel-related microinstructions, because its number indicates the **selected channel**. CHAN register must not be used to store temporary values in arithmetic operations. For more details, refer to [Channel Selection Register - CHAN](#).

### 35.15.8.1.9 Counter Registers: TCR1, TCR2, TPR and TRR

All these registers are 24-bit wide except TPR, which is a 16-bit register. They can be read or written in arithmetic/logical operations, and have special-purpose uses for time base and angle mode operations. For more information about those registers see [Time bases](#) and [EAC - eTPU Angle Counter](#).

### 35.15.8.1.10 General Purpose Registers: A, B, C and D

A, B, C and D are 24-bit general purpose registers, which can be used to store intermediate values and do not have other specific uses with any eTPU feature.

## 35.15.8.2 ALU and Post-ALU Shifter

The ALU executes 24-bit arithmetic and logical operations. ALU's output goes directly to a 1-bit shifter, called post-ALU shifter, so it is possible, for example, to add and shift using only one microinstruction.

In some microinstruction formats, it is not possible to specify the operation executed by ALU. In these cases ALU will always perform addition.

In formats which have the field ALUOP for ALU operation selection, all of them can be performed, including add/subtract using C (carry) flag as ALU's carry-in, bitwise AND/OR/NOT/XOR, and shift/rotate of 2, 4, 8 and 16 bits. See [ALU/MDU Operation Selection](#).

Subtraction, inversion, increment and decrement can be performed by combinations of source inversion and setting ALU's carry-in to 1.

ALU always performs 24-bit operations on its inputs, called **A-source** and **B-source**, and outputs a 24-bit result. 8-, and 16-bit inputs are zero padded to 24 bits. Likewise, ALU 24-bit output is always truncated to the destination register size.

### 35.15.8.2.1 ALU flags

Four flags - Carry, Negative, Overflow, Zero - described below, are related to ALU and post-ALU shift operations. Operation size and shifting affect flags generation logic. Operation size determines the result boundary to be used for flags generation. Operation size is determined by size of sources and destination (see [Flags Sampling Control](#)). For more information about flag generation, see [Flags Sampling Control](#). ALU flags can be used as branch condition (see [Conditional/unconditional branch](#)) or conditional ALU/MDU operation (see [Conditional ALU/MDU Operation Execution](#)).

Field CCS/CCSV in microinstructions can force **no update** of all flags. Not all flags are updated in all ALU operations: Overflow is updated only on addition and absolute value operations, Carry flag is updated in most ALU operations, and only Zero and Negative are updated in all ALU operations.

ALU flags are never updated when microinstruction starts an MDU operation, regardless of CCS/CCSV, but are updated normally afterwards, on ALU operations that are executed in parallel with an ongoing MDU operation (MDU has its own flags).

#### Note

Operation size can be smaller than destination register. For example:  $0xFFFF + 0x0001$  (both 16-bit sources) stores  $0x10000$  in a 24-bit register and sets Zero and Carry flags because operation size is 16 bits.

#### 35.15.8.2.1.1 Carry flag (C)

In an unsigned addition without shifting, Carry flag is the ALU carry from bit 7 to 8, 15 to 16, or 23 to 24 on 8-, 16- and 24-bit operation sizes respectively. In an unsigned subtraction without shifting, Carry Flag represent the sign of ALU's result considering operation size (Carry Flag equal to 1 means a negative result).

Carry flag definition is operation-dependent. The Carry flag in add/subtraction with Post-ALU shift is defined in [Table 35-362](#). Find the definitions for other operations in the following sections.

### 35.15.8.2.1.2 Negative flag (N)

Negative flag indicates the sign of result based on the operation size, regardless of the operation performed, as shown in the following table.

**Table 35-358. Negative (N) flag behavior**

| Operation size | Value                   |
|----------------|-------------------------|
| 8 bits         | $N = \text{result}[7]$  |
| 16 bits        | $N = \text{result}[15]$ |
| 24 bits        | $N = \text{result}[23]$ |

### Note

The N flag may not reflect the sign of the value actually written into the destination register, if it does not have the same size of the operation (see [Flags Sampling Control](#)). This is always the case for registers RAR (14 bits) and CHAN (5 bits).

### 35.15.8.2.1.3 Overflow (V)

Overflow is updated only on addition (with or without carry) and absolute value operations. In signed operations, overflow flag indicates that the result of arithmetic operation (add or subtraction) can not be represented by a word of the size of the operation. Overflow flag behavior for addition is defined in the following table. Overflow flag for Absolute operation is explained in [Absolute Value operation](#). V Flag is calculated using ALU adder output (i.e., it is not affected by 1-bit shift/rotate operations).

**Table 35-359. Overflow flag on addition - V**

| Op. size | Value <sup>1</sup>                                                                                                |
|----------|-------------------------------------------------------------------------------------------------------------------|
| 8 bits   | $(AS[7] \& BS[7] \& !\text{alu\_adder\_output}[7]) \vee (!AS[7] \& !BS[7] \& \text{alu\_adder\_output}[7])$       |
| 16 bits  | $(AS[15] \& BS[15] \& !\text{alu\_adder\_output}[15]) \vee (!AS[15] \& !BS[15] \& \text{alu\_adder\_output}[15])$ |
| 24 bits  | $(AS[23] \& BS[23] \& !\text{alu\_adder\_output}[23]) \vee (!AS[23] \& !BS[23] \& \text{alu\_adder\_output}[23])$ |

1. BS is taken after any inversion by the BINV field, but not added to the carry bit (CIN field)



**NOTE**

For V-flag definition on the absolute operation, see “Absolute Value Operation”.

**35.15.8.2.1.4 Zero flag (Z)**

Zero flag equal to 1 indicates that the result from the ALU, limited to the operation size, is zero, regardless of the operation performed, whether the result is written, or where it is written. It depends on the operation size, as shown in the following table.

**Table 35-360. Zero flag - Z**

| Operation size | Value                                   |
|----------------|-----------------------------------------|
| 8 bits         | $Z = (\text{result}[7:0] == 0x00)$      |
| 16 bits        | $Z = (\text{result}[15:0] == 0x0000)$   |
| 24 bits        | $Z = (\text{result}[23:0] == 0x000000)$ |

**35.15.8.2.2 ALU ADD operation with and without shifting**

ADD operation is selected by ALUOP or ALUOPI fields and when none of them is available in a microinstruction format<sup>33</sup>. Optionally, result can be shifted or rotated right by 1 bit, which is selected by SHF, ALUOP or ALUOPI fields. See [Microinstruction Set](#) for more details. The following table describes how CIN and BINV fields change ADD operation behavior.

**Table 35-361. Types of ADD operations**

| BINV | CIN | Operation (adder output) |
|------|-----|--------------------------|
| 1    | 1   | $AS + BS$                |
| 1    | 0   | $AS + BS + 1$            |
| 0    | 0   | $AS - BS$                |
| 0    | 1   | $AS - BS - 1$            |

ALU adder output can be 1-bit shifted or 1-bit rotated right as follows:

Shift right:

```
if BINV==1
    result [23:0] = adder_output [24:1]
else
    result [23:0] = adder_output [24:1] xor 0x800000
endif
```

Shift left:

33. ALU operations only occur on formats where a destination field is found (T2ABD/T2D).

## Functional description

```
result[23:1] = adder_output[22:0]
result[0]     = 0
```

### Rotate right:

```
case (opsize/CCSV)
8-bit:
    result[6:0] = adder_output[7:1]
    result[7]   = adder_output[0]
    result[23:8] = adder_output[23:8]
16-bit:
    result[14:0] = adder_output[15:1]
    result[15]   = adder_output[0]
    result[23:16] = adder_output[23:16]
24-bit:
    result[22:0] = adder_output[23:1]
    result[23]   = adder_output[0]
```

### Note

Only for the Post-ALU rotate right, the operation size is determined by the field CCSV (see [Flags Sampling Control](#)). For example: if CCSV=00, T4ABS=P (24-bits), T4BBS=A (24 bits), T2ABD=B (24 bits), and ALUOP="Add ROR", then B gets A+P with bits 7:0 rotated, even though the operation size is 24 bits.

The following table describes Carry flag behavior.

**Table 35-362. Carry flag update on ADD operation**

| BINV <sup>1</sup>                                                                            | Op. size | shift/rotate | Value                              |
|----------------------------------------------------------------------------------------------|----------|--------------|------------------------------------|
| 1                                                                                            | 8 bits   | none         | adder carry from bit 7 to bit 8    |
| 1                                                                                            | 16 bits  | none         | adder carry from bit 15 to bit 16  |
| 1                                                                                            | 24 bits  | none         | alu_adder_output[24]               |
| 0                                                                                            | 8 bits   | none         | !adder carry from bit 7 to bit 8   |
| 0                                                                                            | 16 bits  | none         | !adder carry from bit 15 to bit 16 |
| 0                                                                                            | 24 bits  | none         | !alu_adder_output[24]              |
| 0 or 1                                                                                       | 8 bits   | shift left   | alu_adder_output[7]                |
| 0 or 1                                                                                       | 16 bits  | shift left   | alu_adder_output[15]               |
| 0 or 1                                                                                       | 24 bits  | shift left   | alu_adder_output[23]               |
| 0 or 1                                                                                       | x        | shift right  | alu_adder_output[0]                |
| 1                                                                                            | 8 bits   | rotate right | adder carry from bit 7 to bit 8    |
| 1                                                                                            | 16 bits  | rotate right | adder carry from bit 15 to bit 16  |
| 1                                                                                            | 24 bits  | rotate right | alu_adder_output[24]               |
| 0                                                                                            | 8 bits   | rotate right | !adder carry from bit 7 to bit 8   |
| 0                                                                                            | 16 bits  | rotate right | !adder carry from bit 15 to bit 16 |
| 0                                                                                            | 24 bits  | rotate right | !alu_adder_output[24]              |
| except on max-constant generation (see <a href="#">Max Const Generation With T4BBS=111</a> ) |          |              |                                    |

1. BINV has no effect on carry-out when used to code max constant (see [Max Const Generation With T4BBS=111](#)).

Flags N and Z on shift are updated according to the result after shift. Flag V with shift is updated according to the ADD operation only, the same way as without shift.

### 35.15.8.2.3 ADC Operation

ADC operation is selected by the ALUOP field. CIN field is ignored when this operation is selected. The following table describes how BINV change ADC operation behavior.

**Table 35-363. Types of ADC operations**

| BINV | CIN | Operation        |
|------|-----|------------------|
| 1    | x   | AS + BS + C flag |
| 0    | x   | AS - BS - C flag |

Flags behave exactly the same way as for ADD operation without shift/rotate.

### 35.15.8.2.4 Bitwise Operations

Bitwise AND, OR and XOR are selected by ALUOP field. On these operations CIN field is ignored and BINV field inverts (bitwise NOT) BS. C and V Flags are never updated on these operations. The following table describes AND, OR and XOR bitwise operations.

**Table 35-364. Types of Bitwise Operations**

| ALUOP | BINV | Operation  |
|-------|------|------------|
| 10000 | 1    | AS   BS    |
| 10000 | 0    | AS   (~BS) |
| 10001 | 1    | AS ^ BS    |
| 10001 | 0    | AS ^ (~BS) |
| 10010 | 1    | AS & BS    |
| 10010 | 0    | AS & (~BS) |

### 35.15.8.2.5 Set Bit / Clear Bit Operations

These operations set or clear the AS bit determined by BS[4:0]. If the bit number resolves to a value greater than 23, no bit is set or cleared (i.e., result is equal to AS). On these operations CIN field is ignored and BINV field inverts (bitwise NOT) BS. C and V flags are never updated for set/clear bit operations. These operations **override** B-Source size to 8 bits.

set bit (BINV = 1):

## Functional description

```
result = AS | (1 << BS)
```

clear bit (BINV = 1):

```
result = AS & ~(1 << BS)
```

set bit (BINV = 0):

```
result = AS | (1 << (31 - BS))
```

clear bit (BINV = 0):

```
result = AS & ~(1 << (31 - BS))
```

### 35.15.8.2.6 Exchange Bit

Exchange the AS bit determined by BS with C flag. If the bit number resolves to a value greater than 23, no exchange is performed (i.e., result is equal to AS and C flag is not updated). This operation **overrides** BS size to 8 bits. On this operation, CIN field is ignored and BINV field inverts (bitwise NOT) BS. V flag is never updated on exchange bit operation. C flag is always updated, regardless of CCSV, unless BS > 23.

Exchange Bit (BINV=1):

```
if BS <= 23
begin
    temp_C_flag = AS[BS]
    if C_flag == 1
        result = AS | (1 << BS)
    else
        result = AS & ~(1 << BS)
    C_flag = temp_C_flag
end
```

Exchange Bit (BINV = 0):

```
if (31 - BS) <= 23
begin
    temp_C_flag = AS[31 - BS]
    if C_flag == 1
        result = AS | (1 << (31 - BS))
    else
        result = AS & ~(1 << (31 - BS))
    C_flag = temp_C_flag
end
```

### 35.15.8.2.7 Multibit shift/rotate operations

These operations shift or rotate AS by 2, 4, 8 or 16 bits. Size of shift/rotate is determined by BS[1:0]. The following table describes the number of shifted/rotated bits depending on BS[1:0] value.

**Table 35-365. Number of shifted/rotated bits for each BS[1:0] value**

| BS[1:0] | Bits shifted/rotated |
|---------|----------------------|
| 0       | 2                    |
| 1       | 4                    |
| 2       | 8                    |
| 3       | 16                   |

Shift right is a logical operation (i.e., zeros are inserted on left). Multibit shift and rotate operations **overrides** BS size to 8 bits. The shifts and rotate operate on 24 bits, independently of the operation size.

V flag is never updated for multibit shift or rotate operations. Carry flag behavior is described in the following table. CIN is ignored in these operations, but BINV is effective.

**Table 35-366. Carry flag value on multibit shift/rotate operations**

| ALUOP                | BS[1:0] | C flag value <sup>1</sup> |
|----------------------|---------|---------------------------|
| 11001 (shift left)   | 0       | AS[22]                    |
| 11001 (shift left)   | 1       | AS[20]                    |
| 11001 (shift left)   | 2       | AS[16]                    |
| 11001 (shift left)   | 3       | AS[8]                     |
| 11010 (shift right)  | 0       | AS[1]                     |
| 11010 (shift right)  | 1       | AS[3]                     |
| 11010 (shift right)  | 2       | AS[7]                     |
| 11010 (shift right)  | 3       | AS[15]                    |
| 11011 (rotate right) | 0       | AS[2]                     |
| 11011 (rotate right) | 1       | AS[4]                     |
| 11011 (rotate right) | 2       | AS[8]                     |
| 11011 (rotate right) | 3       | AS[16]                    |

1. CCS/CCSV can disable flag update on multibit shift/rotate, but the specified flag size in CCSV is ignored for the C flag.

### 35.15.8.2.8 Absolute Value operation

Absolute Value operation is selected by ALUOP field. On this operation, AS is interpreted as a signed number and its absolute value is the result. V and N flags are updated with the result signal determined by the operation size. AS bit 23 after size override and sign extension (if any, see [A-Source Size Override](#)), regardless of A-source register size, is used to check the operand signal and is copied to C-flag. Note that if AS is 8-bit or 16-bit, its sign is taken into account and copied to C only if sign-extension is performed. The following table summarizes flag updating for Absolute Value operation.

This operation is independent of B-source. Instruction fields T4BBS, BINV and CINV are ignored in this operation. The Absolute Value operation size is the minor between A-source size and Destination size.

**Table 35-367. ALU flags in Absolute Value operation**

| operation size | V, N <sup>1</sup> | C      | Z                     |
|----------------|-------------------|--------|-----------------------|
| 8              | alu_output[[7]    | AS[23] | alu_output[7:0] == 0  |
| 16             | alu_output[15]    |        | alu_output[15:0] == 0 |
| 23             | alu_output[23]    |        | alu_output[23:0] == 0 |

1. V, N can be 1 on 8- and 16-bit Absolute Value, because the operand sign is always taken from bit 23. V, N can also be 1 in 23-bit Absolute Value (or 8-bit and 16-bit with sign extension), if the operand is 0x800000 (0x80, 0x8000).

### 35.15.8.3 MAC and Divide Unit (MDU)

MDU is an autonomous resource in the microengine which can carry out sequential multiply, multiply-accumulate, fractional multiplication and divide operations, selected through the microinstruction fields ALUOP or ALUOPI. The unit supports signed and unsigned multiply and fractional multiplication of any combination of 8, 16 or 24 bit operands <sup>34</sup>, and also signed and unsigned 24-bit multiply-accumulate. Divide operation is unsigned, and both operands are always 24-bit wide.

Depending on the size of operands and the type of operation, MDU can take more than one microcycle to execute the operation, but microengine continues to execute microinstructions in parallel. When the microcode issues an END command, any MDU executing operation terminate immediately and is left incomplete. When selecting an operation that uses MDU, the result is always placed in MACH and MACL registers, and the register selected as destination does not have its value changed ([Selecting sources and destination](#)). During calculations, MACH and MACL holds temporary values and should not be written, otherwise the result is unspecified. One must not start an MDU operation while MDU is already busy: the result is unspecified for both the ongoing operation and the started one.

MDU Operations update its own set of 5 flags, described in [MDU Flags](#). MDU operations never update C, N, V and Z flags. CIN and BINV microinstruction fields affect MDU operations according to the following table.

**Table 35-368. CIN and BINV with MDU Operations**

| B-source operand | BINV | CIN | Op. Performed |
|------------------|------|-----|---------------|
| signed           | 1    | 1   | AS mdu_op BS  |

*Table continues on the next page...*

34. There is no distinct selection of 24-bit fractional multiplication, for it works exactly as a 24-bit ordinary multiplication.

**Table 35-368. CIN and BINV with MDU Operations (continued)**

| B-source operand      | BINV | CIN | Op. Performed    |
|-----------------------|------|-----|------------------|
|                       | 0    | 0   | AS mdu_op (-BS)  |
|                       | 1    | 0   | reserved         |
|                       | 0    | 1   | reserved         |
| unsigned <sup>1</sup> | 1    | 1   | AS mdu_op BS     |
|                       | 1    | 0   | AS mdu_op (BS+1) |
|                       | 0    | x   | reserved         |

1. includes the B-source (unsigned) in fmults (signed) operations.

### 35.15.8.3.1 Multiply and Multiply-Accumulate Operation Length

MDU needs 2 sources, A source and B source, to perform an operation. The time needed to perform a multiply or multiply-accumulate is:

- on 24-bit x 8 bit multiplies: 2 microcycles (one start-MDU plus one execution microcycle)
- on 24-bit x 16 bit multiplies: 3 microcycles (one start-MDU plus two execution microcycles)
- on 24 bit x 24 bit multiplies/macs: 4 microcycles (one start-MDU plus three execution microcycles)

An internal pipeline in MDU allows multiply-accumulate (or even non-fractional multiply) operations to start one microinstruction before a multiplication or multiply-accumulate (signed or unsigned) has been completed (e.g., one can start one multiply or multiply-accumulate once every three microinstructions). However, by doing that it is not possible to read the result in MACH and MACL (although the MDU flags can be tested), so this is intended to be used in a multiply-accumulate sequence. It is also allowed to mix different sizes in multiply/mac sequences.

Multiply-accumulate operations are similar to multiply operations, except that the contents of MACH and MACL registers are added to the multiplication result.

When multiply or multiply accumulate operations finish, MACL and MACH hold the least and the most significant 24-bit words, respectively.

### 35.15.8.3.2 Divide Operation Length

The divide operation is always unsigned. The division completes in 13 microcycles, meaning that after the start divide microinstruction, one has to wait for 12 microcycles and then read the result and the remainder in MACH and MACL registers. During the 12 execution microcycles, microengine can execute microinstructions unrelated to the MDU.

### 35.15.8.3.3 Signed Multiplication (mults)

MDU signed multiplication is defined as follows:

```
(signed) MACH,MACL = (signed) AS * (signed) BS
```

MC and MV flags are reset. MZ is set if result is 0, resets otherwise. MN is set if result is negative.

### 35.15.8.3.4 Unsigned Multiplication (multu)

MDU unsigned multiplication is defined as follows:

```
(unsigned) MACH|MACL = (unsigned) AS * (unsigned) BS
```

MC and MV flags are reset. MZ is set if result is 0, resets otherwise. MN is a copy of the most significant bit of result.

### 35.15.8.3.5 Signed Multiply-Accumulate (macs)

MDU signed multiply-accumulate is defined as follows:

```
(signed/unsigned) {MACH,MACL} += (signed) AS * (signed) BS
```

MC is not altered.

MV is set if result can not be represented by a 48-bit signed number. MACS never resets MV flag: it is left as is if no overflow occurs, or set it otherwise. This allows checking the overflow flag only once at the end of a series of multiply-accumulate operations in a scalar product calculation.

```
if (({MACH,MACL} += AS * BS < -247) || ({MACH,MACL} += AS * BS > 247 - 1))
    MV = 1
```

MZ is set if result is 0, resets otherwise. MN is a copy of the most significant bit of result.

Note that only 24-bit multiply-accumulate is available.



### 35.15.8.3.6 Unsigned Multiply-Accumulate (macu)

MDU Unsigned Multiply-Accumulate is defined as follows:

```
(signed/unsigned) {MACH,MACL} += (unsigned) AS * (unsigned) BS
```

MC is set if result can not be represented by a 48-bit unsigned non-negative number. MACU never resets MC flag: MC flag is left as is if no carry occurs, or set otherwise. This allows checking the carry flag only once at the end of a series of multiply-accumulate operations in a scalar product calculation.

```
if (({MACH,MACL} += AS * BS < 0) || ({MACH,MACL} += AS * BS > 248 - 1))
    MC = 1
```

MV is not altered.

MZ is set if result is 0, resets otherwise. MN is a copy of the most significant bit of result.

Note that only 24-bit multiply-accumulate is available.

### 35.15.8.3.7 Signed Fractional Multiplication (fmults)

MDU Signed Fractional Multiplication takes the B-Source as an **unsigned** 8- or 16-bit<sup>35</sup> fraction between 0 and  $(2^8 - 1)/2^8$  (inclusive) for the 8-bit operation, or between 0 and  $(2^{16}-1)/2^{16}$  (inclusive) for the 16-bit operation. Only A-Source is taken as a signed number. The value of B-Source is considered the unsigned numerator of a fraction with denominator  $2^8$  or  $2^{16}$  for the 8- and 16-bit operations, respectively.

The integer part of the result is stored in MACH, and the fractional part in MACL. The result is signed, so that the concatenation of MACH and MACL form a 48-bit fixed point number with a 24-bit integer part, both for 8- and 16-bit operations. To calculate the unsigned numerator of the fractional part (with denominator  $2^{24}$ ) of the result, one must take the absolute value of MACL considering the signal of the result (not MACL alone), i.e.: if flag MN=1, invert MACL and add 1.

MDU flags are updated in the same way as in the Signed Multiplication.

### 35.15.8.3.8 Unsigned Fractional Multiplication (fmultu)

MDU Unsigned Fractional Multiplication takes both A-Source and B-Source as unsigned operands. B-Source is taken as an 8- or 16-bit<sup>36</sup> fraction between 0 and  $(2^8 - 1)/2^8$  (inclusive) for the 8-bit operation, or between 0 and  $(2^{16}-1)/2^{16}$  (inclusive) for the 16-bit operation. The value of B-Source is considered the numerator of a fraction with denominator  $2^8$  or  $2^{16}$  for the 8- and 16-bit operations, respectively.

35. 24-bit fractional signed/unsigned multiplication is equivalent to mults/multu.

36. 24-bit fractional signed/unsigned multiplication is equivalent to mults/multu.

The integer part of the result is stored in MACH, and the fractional part in MACL. The fractional part in MACL is the numerator of a fraction with denominator  $2^{24}$ . The concatenation of MACH and MACL form a 48-bit fixed point number with a 24-bit integer part, both for 8- and 16-bit operations.

MDU flags are updated in the same way as in the Unsigned Multiplication.

### **35.15.8.3.9 Unsigned Divide (div)**

At the end of a divide operation MACL holds the result of the division, taking A-source as numerator and B-source as denominator, while MACH holds the remainder. If a divide by 0 is executed, MACL holds the maximum unsigned number (0xFFFFFFFF) as result and flag MV is set to indicate division by 0 (otherwise reset). The contents of MACH become indetermined.

MC flag is always reset.

MZ flag is set if MACL equals 0, and reset otherwise.

MN receives a copy of MACH bit 23 (msb from the remainder).

Note that signed division is not available.

### **35.15.8.3.10 MDU Flags**

MDU has its own flags to indicate the result and status of an MDU operation. They are: MC, MZ, MV, MN and MB. All MDU flags are updated with the final result at the end of the operation, and do not change until the next operation finishes. Therefore it is possible to start a new MDU operation and test the flags of the previous one in parallel, except for mult/mac with 8-bit operand (takes only 1 microcycle).

#### **35.15.8.3.10.1 MDU Negative Flag - MN**

MN flag is always a copy of MACH bit 23 at the end of the operation, either in signed or unsigned ones. Note that MACH holds the rest of a division operation, which is always unsigned.

#### **35.15.8.3.10.2 MDU Carry Flag - MC**

MDU carry flag indicates if the result cannot be represented by a 48-bit number, in Signed and Unsigned Multiply Accumulates. It is reset in the other operations.

### 35.15.8.3.10.3 MDU Zero Flag - MZ

In multiply and multiply-accumulate operations, MDU zero flag is asserted if MACH and MACL are equal to zero at the end of an operation. In divide operations, zero flag is asserted if MACL (result) is equal to 0.

### 35.15.8.3.10.4 MDU Overflow Flag - MV

In multiply operations, MV flag is negated and keeps negated in the end, because the result of a multiplication can always fit in a 48-bit result (MACH and MACL concatenated). In a multiply-accumulate operation, MV is asserted if the result size is wider than 48 bits. MV flag work in both signed and unsigned operations.

In divide operations it is only asserted if a divide-by-zero operation was executed.

### 35.15.8.3.10.5 MDU Busy Flag (MB)

MB tests as true at the next microinstruction after the MDU start operation, and as false at the last microcycle of any MDU operation execution.

## 35.15.8.4 Branch conditions

Microengine allows conditional branch. There are five sets of flags that can be tested in a conditional branch: ALU flags, MDU flags, P flags, Channel flags, and Semaphore flag (flag SMLCK).

When a thread starts to be executed, the values in MDU and ALU flags are not initialized. ALU flags are described in [ALU flags](#), MDU flags are described in [MDU Flags](#). MDU and ALU flags are updated during execution of microinstructions.

P flags are actually the upper byte of P register, which optionally can work as user defined flags (see [P Register](#)).

Channel flags Flags0, Flag1, MRLA, MRLB, TDLA, TDLB, PSS, PRSS and PSTO are obtained from the selected channel (value in CHAN register), while channel flags, LSR, FM[0] and FM[1] are selected by the serviced channel, regardless of the CHAN value<sup>37</sup>.

Flags TDLA/B, MRLA/B, LSR, FM and PSS, are sampled at the beginning of a thread. Flag PSS does not change during its execution while CHAN register is not written. When a write in CHAN register is performed, all flags except LSR and FM are updated according to the channel specified by CHAN value. Flags MRLA/B and TDLA/B are reset when their respective latches in channel are cleared by microcode.

---

37. Serviced channel does not change during execution of a thread, and it is the channel that requested a service (initial value of CHAN register when a thread starts).

**Table 35-369. Channel flags as branch condition**

| Flag  | Description                                | Service or selected channel                                                                                                                                                                                                                     |
|-------|--------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Flag0 | State Resolution flag                      | reflects the selected channel (CHAN)                                                                                                                                                                                                            |
| Flag1 | State Resolution flag                      | reflects the selected channel (CHAN)                                                                                                                                                                                                            |
| MRLA  | Match A Recognition Latch                  | these flags reflect the selected channel (CHAN)see <a href="#">MRLA/B - Match Recognition Latches</a> and <a href="#">TDLA/B - Transition Detect Latches</a> for more information.                                                              |
| MRLB  | Match B Recognition Latch                  |                                                                                                                                                                                                                                                 |
| TDLA  | Transition A Detection Latch               |                                                                                                                                                                                                                                                 |
| TDLB  | Transition B Detection Latch               |                                                                                                                                                                                                                                                 |
| LSR   | Link Service Request                       | reflects the serviced channel.                                                                                                                                                                                                                  |
| PRSS  | Input Pin state sampled at service request | reflects the selected channel (CHAN). Does not change if CHAN is not changed.                                                                                                                                                                   |
| PSS   | Sampled Input Pin State                    | reflects the selected channel (CHAN). Does not change if CHAN is not changed (see <a href="#">Pin Control Registers</a> ).                                                                                                                      |
| PSTI  | Current Input Pin State.                   | reflects the selected channel (CHAN). Changes any time.                                                                                                                                                                                         |
| PSTO  | Current Output Pin State                   | reflects the selected channel (CHAN). Changes any time.                                                                                                                                                                                         |
| FM    | Function Mode Bits                         | reflects the Function Mode for serviced channel (see <a href="#">eTPU Channel n Status Control Register for Engine 1</a> (eTPU_CCCR_CnSCR_ENG1) and <a href="#">eTPU Channel n Status Control Register for Engine 2</a> (eTPU_CCCR_CnSCR_ENG2)) |

Semaphore condition SMLCK always indicates if a semaphore is locked for the Engine, resolving as false before any lock attempt. For each trial, the SMLCK flag is updated. The SMLCK value set in one thread is not meaningful to the other. After a free, the SMLCK condition tests as false until a new lock attempt on the same thread.

Branch conditions are selected through instruction fields BCC and BCF (see [Conditional/unconditional branch](#)).

### 35.15.9 Microinstruction Set

Each microinstruction can execute up to 3 microoperations in parallel. Microinstructions are grouped into formats, and there are four types of microoperations:

- ALU/MDU Operations
- SDM Operations
- Channel Configuration/Control Operations
- Flow Control Operations

Each microinstruction format is defined by a set of microinstruction fields, which determine the operations, each belonging to one of the groups above (there may be several in one group). Complete microinstruction formats are shown in [Microinstruction formats](#).

Parallelism conflicts may arise when two operations are executed in the same microinstruction. These situations are explained in [Microinstruction Parallelism Issues](#).

### 35.15.9.1 SDM Microoperations

The access to SDM is made by providing an address and a register to perform a data transfer, except semaphore operations, which are also classified in the SDM group. Only P and DIOB registers can exchange data with SDM. Microengine always addresses SDM in 32-bit boundaries, for 8, 24, or 32 bit wide data.

Direction is determined by the field RW in all addressing modes: RW=0 selects read and RW=1 selects write.

#### 35.15.9.1.1 SDM Addressing Modes

There are 4 addressing modes in eTPU:

- Absolute
- Selected Channel Relative
- Indirect
- Engine Relative

The addressing modes Absolute and Selected Channel Relative use immediate bits to form the physical address of SDM, which is identified in microinstruction as a field called AID. AID field can be 3-, 7-, or 8-bit wide depending on the addressing mode.

##### 35.15.9.1.1.1 Absolute Addressing Mode

In Absolute addressing mode, the address range is 256 parameters, addressed by field AID, which in this mode is 8-bit wide. These parameters are located in SDM addresses from 0 to 255.

```
physical_address = AID[7:0]
```

### 35.15.9.1.1.2 Selected Channel Relative Addressing Mode

In Selected Channel Relative addressing mode, only the first 8 (with 3-bit AID) or 128 (with 7-bit AID) parameters of the selected channel are accessible, depending on the microinstruction format. Physical address is calculated using the channel parameter base address that is specified in the CPBA field of [eTPU Channel n Configuration Register for Engine 1 \(eTPU\\_CCCR\\_CnCR\\_ENG1\)](#) and [eTPU Channel n Configuration Register for Engine 2 \(eTPU\\_CCCR\\_CnCR\\_ENG2\)](#). AID field is added to channel parameter base address to compose the physical address. The address equation is:

```
physical_address = selected_channel_parameter_base_address + AID[6:0], or
physical_address = selected_channel_parameter_base_address + AID[2:0]
```

### 35.15.9.1.1.3 Indirect Addressing Mode

In Indirect Addressing mode the physical address is taken from DIOB register. Only DIOB bits 14 to 2 are relevant. Since the SDM word address is shifted two bits up in DIOB, its contents hold the same parameter address value used by Host. The address equation for SDM access is:

```
physical_address = DIOB[13:2], or
physical_address = (truncated to 12 bits) (DIOB / 4)
```

Indirect addressing mode is used to access the Shared Subsystem Area when `eTPU_CCCR_CxCR_ENGn[SSMAE]=1` and DIOB addresses the 16 Kbytes just above the SDM range (DIOB bit 14 = 1). See [Shared Subsystems support](#) for more details. If `SSMAE=0`, accesses with DIOB bit 14 = 1 hit the same SDM addresses as with bit 14 = 0, providing full backwards compatibility with respect to DIOB contents.

Indirect addressing mode can have post-increment or pre-decrement on DIOB, allowing stack operations. See [DIOB stack operation](#) for more information.

### 35.15.9.1.1.4 Engine Relative Addressing Mode

In Engine Relative Addressing mode the physical address is the concatenation of the `eTPU_SCR_ECR_ENGn[ERBA]` field (see [eTPU Engine 1 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG1\)](#) and [eTPU Engine 2 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG2\)](#)) with the 7-bit AID instruction field. This allows the same function microcode, when running on distinct engines, to access different address spaces, global to the engine only.

The SDM physical word address in Engine Relative Addressing mode is calculated as follows:

```
physical address = (eTPU_SCR_ECR_ENGn[ERBA] << 7) + AID[6:0]
```

### 35.15.9.1.2 SDM source/destination registers

When performing an SDM operation, only DIOB or P can be used as data source or destination. P is 32-bit wide, and DIOB is 24-bit wide. Microinstruction field P/D (1 bit) is used to choose between P and DIOB as data the source or destination. When the P/D field is not available in microinstructions that support SDM access, the source/destination is P.

**Table 35-370. SDM source/destination Register Selection**

| P/D | Meaning     |
|-----|-------------|
| 0   | P access    |
| 1   | DIOB access |

### 35.15.9.1.3 SDM Operation Size

When using DIOB register to perform SDM data transfers, the operation size is always 24-bit wide (lower 24 bits of SDM). When using P register, the operation size can be 8-, 24- or 32-bit wide, which is controlled by microcode RSIZ field (2 bits). RSIZ meaning is shown in the following table.

**Table 35-371. SDM P access Size**

| RSIZ | P access                                                              | DIOB access             |
|------|-----------------------------------------------------------------------|-------------------------|
| 00   | full 32-bit access (i.e. P[31:0]=SDM[addr] [31:0])                    | RESERVED                |
| 01   | only upper 8 bits are transferred (i.e. P[31:24] = SDM[addr] [31:24]) | RESERVED                |
| 10   | only lower 24 bits are transferred (i.e. P[23:0] = SDM[addr] [23:0])  | DIOB = SDM[addr] [23:0] |
| 11   | RESERVED                                                              | RESERVED                |

RSIZ is not available in all microinstructions that support SDM access. In microinstructions where RSIZ field is not available, SDM access will be 24 bits by default.

When performing a Zero SDM write operation (see [Zero SDM Operation](#)), RSIZ defines the size of operation regardless of the P/D field ([SDM source/destination registers](#)).

### 35.15.9.1.4 SDM access direction

RW field defines the direction of the access in the SDM. The access direction is summarized in the following table.

**Table 35-372. SDM access direction**

| R/W | Meaning                                      |
|-----|----------------------------------------------|
| 0   | read SDM parameter into P or DIOB registers  |
| 1   | write SDM parameter from P or DIOB registers |

### 35.15.9.1.5 Zero SDM Operation

Zero SDM operation is controlled by microcode field ZRO (1bit). When ZRO field is 0, the data portion written in SDM or in P/DIOB (SDM read) registers will always be 0x0. When performing a Zero SDM write operation, the RSIZ is relevant regardless of the P/D field (usually RSIZ is meaningful only for P/D = 0), which means that zero SDM write operation can be performed with 32, 24 or 8 bits according to SDM operation size. These conditions are summarized in the following table.

**Table 35-373. Zero SDM Operation**

| ZRO | RW | P/D | Meaning                                                                                       |
|-----|----|-----|-----------------------------------------------------------------------------------------------|
| 0   | 0  | 0   | Clear P register. Size is determined by RSIZ field. See <a href="#">SDM Operation Size</a>    |
| 0   | 0  | 1   | Clear DIOB (all 24 bits), independently of RSIZ                                               |
| 0   | 1  | x   | Clear SDM parameter. Size is determined by RSIZ field. See <a href="#">SDM Operation Size</a> |
| 1   | RW | P/D | Regular SDM operation                                                                         |

### Note

When field STC is present, STC=11 also disables Zero SDM operation (see [Table 35-374](#)). The conflicts with DIOB operations (see [DIOB stack operation](#)) and ALU operations are resolved like a normal SDM operation (see [Table 35-415](#)).

Zero SDM read operations do check memory errors on the read data, however discarded. Zero SDM write operations normally update the ECC of the written data.

### 35.15.9.1.6 DIOB stack operation

SDM Indirect Addressing Mode (see [Indirect Addressing Mode](#)) is used if STC field (2 bits) exists in the microinstruction, controlling automatic increment/decrement of DIOB register, as shown in the following table, thus allowing stack operations. DIOB is incremented and decremented in word addresses, only from bits 15 downto 2, i.e.: the bits 23 to 16 and 1 to 0 are left untouched by STC pre-decrement and post-increment.



**Table 35-374. DIOB Post-Increment / Pre-Decrement - STC**

| STC | Meaning                                |
|-----|----------------------------------------|
| 00  | Post-Increment of DIOB                 |
| 01  | Pre-Decrement of DIOB                  |
| 10  | No Increment/Decrement (normal access) |
| 11  | No SDM Access <sup>1</sup>             |

1. Also disables Zero SDM operation.

### 35.15.9.1.7 Semaphore Operations

Semaphore lock and free operations are available through eTPU microcode. For more information about semaphores see [Hardware Semaphores](#). There are 2 microinstruction fields that control semaphore operations: FL (1 bit) and SMPR (2 bits). Serviced channel sees 4 semaphores, selected by field SMPR.

**Table 35-375. Semaphore Operations Fields**

| Field | Meaning                                   |
|-------|-------------------------------------------|
| FL    | 0 = free semaphore,<br>1 = lock semaphore |
| SMPR  | semaphore number selector                 |

When freeing a semaphore, the field SMPR has no meaning. This is because only one semaphore can be locked at a time by each engine, so when freeing a semaphore it is not necessary to specify its number.

#### Note

If microcode tries to lock a semaphore already locked for the same Engine, the semaphore continues locked for the Engine and the SMLCK branch condition resolves as true.

### 35.15.9.2 ALU/MDU Operations

ALU/MDU microoperations mostly comprises two sources, one destination and one operation. The operation is generally selected through fields ALUOP, ALUOPI or SHF. In formats where there is no operation selection field (ALUOP, ALUOPI or SHF), the operation performed is always addition; however, it is possible to perform subtraction, increment or decrement using fields BINV (see [B-Source Inversion](#)) and CIN (see [Carry-in Control](#)).

### 35.15.9.2.1 Source and Destination Register Set Selection

Microcode field T4ABS allows selection of a source from either one of two register sets, shown in [Table 35-379](#). The same applies to T2ABD, used for ALU destination selection with other two register sets, as shown in [Table 35-380](#). Microinstruction fields ABSE and ABDE control the register set selection for source and destination, respectively, when available at the format. In formats without ABSE/ABDE, the field T4BBS determines the register sets used by T2ABD and T4ABS, as shown in [Microinstructions without fields ABSE and ABDE](#).

#### 35.15.9.2.1.1 Microinstructions With Fields ABSE and ABDE

In microinstructions where ABSE and ABDE fields are available (1 bit each), ABSE controls register set selection for T4ABS and ABDE controls register set selection for T2ABD. The following table shows the meaning of values for ABSE and ABDE fields.

**Table 35-376. Register Set Selection by ABSE or ABDE**

| ABSE or ABDE | Register Set Selected |
|--------------|-----------------------|
| 0            | second                |
| 1            | first                 |

#### 35.15.9.2.1.2 Microinstructions without fields ABSE and ABDE

When ABSE and ABDE are not available in a microinstruction format, register set is selected by T4BBS field. The following table explains how to select a register set either for T4ABS and T2ABD.

**Table 35-377. Register set selection by T4BBS without ABSE and ABDE**

| T4BBS             | Register set for T2ABD | Register set for T4ABS |
|-------------------|------------------------|------------------------|
| 0xx               | first                  | first                  |
| 100               | second                 | second                 |
| 101               | second                 | first                  |
| 110               | first                  | second                 |
| 111               | first                  | first                  |
| none <sup>1</sup> | first                  | first                  |

1. Refers to operations with immediate data as B-source, without ABSE and ABDE.

### 35.15.9.2.2 Selecting sources and destination

All ALU/MDU operations need 2 sources (called AS and BS) and 1 destination (called AD), except for some of those that use immediate data (see [Operations With Immediate Data](#)). Fields T4ABS (4 bits), ABSE (1 bit), T4BBS (3 bits) select sources, while T2ABD (4 bits) and ABDE (1 bit) select the destination. When MDU is used (multiply/divide), T2ABD destination selection is ignored and results are stored in MACH and MACL (see [MAC and Divide Unit \(MDU\)](#)). ABSE and ABDE are not available in some microinstruction formats that support ALU/MDU operations. However, in all formats where ABSE is available, ABDE is also available and vice-versa. The existence of ABSE/ABDE fields changes the meaning of T4BBS field, as shown in the table below. On instructions with immediate data, it is used as B-source (see [Operations With Immediate Data](#)).

All sources and destinations have a size associated to it, and these sizes are used to select flag sample position (see [Flags Sampling Control](#)). Sizes can be 8, 16 or 24 bits. Registers that are not exactly of one of these sizes are treated as the immediately upper size (e.g., CHAN[4:0] is an 8-bit source). See [Flags Sampling Control](#) for more information.

Some parallelism issues arise when selecting P, DIOB, ERTA or ERTB as destination registers, since they can be modified by other microoperations in the same microinstruction (see [Microinstruction Parallelism Issues](#) for details).

**Table 35-378. B source selection - T4BBS**

| T4BBS | Meaning in microinstruction formats with ABSE/ABDE                                         | Meaning in microinstruction formats without ABSE/ABDE <sup>1</sup> |
|-------|--------------------------------------------------------------------------------------------|--------------------------------------------------------------------|
| 000   |                                                                                            | BS[23:0] = P[23:0]                                                 |
| 001   |                                                                                            | BS[23:0] = A[23:0]                                                 |
| 010   |                                                                                            | BS[23:0] = SR[23:0]                                                |
| 011   |                                                                                            | BS[23:0] = DIOB[23:0]                                              |
| 100   | reserved                                                                                   | BS = 0                                                             |
| 101   | reserved                                                                                   | BS = 0                                                             |
| 110   | reserved                                                                                   | BS = 0                                                             |
| 111   | BS=0, or Max const., if CIN=0 and BINV=0 (see <a href="#">Generating "max" constant</a> ). |                                                                    |

1. T4BBS also selects A-source and destination register set in this case, according to [Table 1](#).

T4ABS selects one source from 2 register sets, shown in the following table. ABSE and T4BBS control which set T4ABS field uses to select the source. For more information about how to select a register set for T4ABS and T2ABD see [Source and Destination Register Set Selection](#). All sources are zero-filled to 24 bits, unless sign-extension is specified (see [A-Source Size Override](#)).

**Table 35-379. A source selection - T4ABS**

| T4ABS | First register set    |      | Second register set                  |      |
|-------|-----------------------|------|--------------------------------------|------|
|       | Selected register     | Size | Selected Register                    | Size |
| 0000  | AS[7:0]=P[7:0]        | 8    | AS[7:0]=0                            | 8    |
| 0001  | AS[7:0]=P[15:8]       | 8    | AS[23:0]=C[23:0]                     | 24   |
| 0010  | AS[7:0]=P[31:24]      | 8    | AS[15:0] = TPR[15:0]                 | 16   |
| 0011  | AS[23:0] = ERTB[23:0] | 24   | AS[23:0] = B[23:0]                   | 24   |
| 0100  | AS[23:0] = D[23:0]    | 24   | AS[23:0] = TRR[23:0]                 | 24   |
| 0101  | AS[15:0] = P[15:0]    | 16   | AS[7:0] = 0, read_match <sup>1</sup> | 8    |
| 0110  | AS[15:0] = P[31:16]   | 16   | AS[13:0] = RAR[13:0]                 | 16   |
| 0111  | AS[7:0] = P[23:16]    | 8    | AS[23:0] = MACH[23:0]                | 24   |
| 1000  | AS[23:0] = P[23:0]    | 24   | AS[23:0] = MACL[23:0]                | 24   |
| 1001  | AS[23:0] = A[23:0]    | 24   | AS[4:0]=CHAN[4:0]                    | 8    |
| 1010  | AS[23:0] = SR[23:0]   | 24   | AS[14 :2] = CHAN_BASE <sup>2</sup>   | 16   |
| 1011  | AS[23:0] = DIOB[23:0] | 24   | AS[13:0] = ENGINE_BASE <sup>3</sup>  | 16   |
| 1100  | AS[23:0] = TCR1[23:0] | 24   | Reserved                             | -    |
| 1101  | AS[23:0] = TCR2[23:0] | 24   | Reserved                             | -    |
| 1110  | AS[23:0] = ERTA[23:0] | 24   | Reserved                             | -    |
| 1111  | AS[23:0] = 0          | 24   | Reserved                             | -    |

1. T4ABS=0101 with second register set also reads MatchA/B registers into ERTA/B (see [Special T4ABS Source Operation: Read Match Registers](#)).
2. CHAN\_BASE is the selected channel's base SDM address in channel relative address mode (see [CHAN\\_BASE as a source](#)).
3. ENGINE\_BASE is the eTPU\_SCR\_ECR\_ENGn[ERBA] field shifted left 9 positions.

T2ABD selects the destination from 1 of 2 register sets, shown in the following table. ABDE and T4BBS control which set T2ABD field uses to select the destination.

**Table 35-380. Destination selection - T2ABD**

| T2ABD | First register set                 |      | Second register set   |      |
|-------|------------------------------------|------|-----------------------|------|
|       | Selected register                  | Size | Selected Register     | Size |
| 0000  | A[23:0] = AD[23:0]                 | 24   | C[23:0] = AD[23:0]    | 24   |
| 0001  | SR[23:0] = AD[23:0]                | 24   | LINK[7:0] = AD[7:0]   | 8    |
| 0010  | ERTA[23:0] = AD[23:0] <sup>1</sup> | 24   | TPR[15:0] = AD[15:0]  | 16   |
| 0011  | ERTB[23:0] = AD[23:0] <sup>2</sup> | 24   | B[23:0] = AD[23:0]    | 24   |
| 0100  | DIOB[23:0] = AD[23:0]              | 24   | CHAN[4:0] = AD[4:0]   | 8    |
| 0101  | P[15:0] = AD[15:0]                 | 16   | D[23:0] = AD[23:0]    | 24   |
| 0110  | P[31:16] = AD[15:0]                | 16   | RAR[12:0] = AD[12:0]  | 16   |
| 0111  | P[23:0] = AD[23:0]                 | 24   | MACH[23:0] = AD[23:0] | 24   |
| 1000  | TCR1[23:0] = AD[23:0]              | 24   | MACL[23:0] = AD[23:0] | 24   |
| 1001  | TCR2[23:0] = AD[23:0]              | 24   | Reserved              | -    |
| 1010  | P[31:24] = AD[7:0]                 | 8    | Reserved              | -    |

Table continues on the next page...

**Table 35-380. Destination selection - T2ABD (continued)**

| T2ABD | First register set                   |      | Second register set |      |
|-------|--------------------------------------|------|---------------------|------|
|       | Selected register                    | Size | Selected Register   | Size |
| 1011  | P[23:16] = AD[7:0]                   | 8    | Reserved            | -    |
| 1100  | P[15:8] = AD[7:0]                    | 8    | Reserved            | -    |
| 1101  | P[7:0] = AD[7:0]                     | 8    | Reserved            | -    |
| 1110  | TRR[23:0] = AD[23:0]                 | 24   | Reserved            | -    |
| 1111  | no destination selected <sup>3</sup> | 24   | Reserved            | -    |

1. T2ABD=0010 with first register set also writes to MatchA or UDCM registers of the selected channel if field ERWA=0 (see [Write Channel Match and UDCM Registers](#)).
2. T2ABD=0011 with first register set also writes to MatchB register of the selected channel if field ERWB=0.
3. if no destination is selected, ALU flags are updated, although the result is lost.

### 35.15.9.2.2.1 Max Const Generation With T4BBS=111

When T4BBS = 111, BINV = 0, and CIN = 0, the value assigned to BS will be 0x800000, and not 0x0 as expected. See [Generating "max" constant](#) for detailed explanation about this.

### 35.15.9.2.2.2 Special T4ABS Source Operation: Read Match Registers

When T4ABS = 0101 and the source for T4ABS is selected from the second register set, the constant 0x00 is used as AS (8-bit size) and the following register transfer is performed in parallel as well: match registers of the selected channel (value in CHAN register) are copied to ERTA/ERTB registers, where ERTA receives the value of MatchA register and ERTB receives the value of MatchB register (see [ER - Event Registers](#)). Note that ALU destination can still be chosen by T2ABD in parallel. When ERTA or ERTB is selected by T2ABD, a parallelism issue arises (see [ALU Operations and Read Match Registers](#)).

### 35.15.9.2.2.3 CHAN\_BASE as a source

Each channel has a parameter base address in SDM, which is configured in the CPBA field of [eTPU Channel n Configuration Register for Engine 1](#) (eTPU\_CCCR\_CnCR\_ENG1) and [eTPU Channel n Configuration Register for Engine 2](#) (eTPU\_CCCR\_CnCR\_ENG2). CHAN\_BASE, which represents a parameter address (CPBA\*2), can be used as A-source using T4ABS=1010 when T4ABS selects a source from the second register set. In this case, CHAN\_BASE is loaded into AS[13:2] to form the byte address (AS[23:14]=0, AS[1:0]=0). For example, in Indirect addressing mode, where the destination register is DIOB, CHAN\_BASE is loaded into DIOB[13:2], which is the parameter address, and DIOB[13:0] represents the byte address. CHAN\_BASE is the base address of the selected channel (given by CHAN register).

### 35.15.9.2.3 Flags Sampling Control

This section explains how the flags Z (zero), C (carry), N (negative) and V (overflow) are updated in an ALU operation. When there are post-ALU shift operations, the ALU Carry Out is not directly sampled in Carry flag, but passed to the post-ALU shifter (see [Shift Operations](#)). Since the size of source operands in ALU operations is variable, flags can be sampled as an operation of 8, 16 or 24 bits wide. The operation size selection is automatic, based on defined sizes of sources and destination, using the equation:

```
operation_size = minor(size_of(destination), greater(size_of(A-Source),
size_of(B-Source)))
```

Operation size can also be shown with the following table:

**Table 35-381. Operation Size Determination**

| A Source      | B Source     | Destination | Operation Size |
|---------------|--------------|-------------|----------------|
| x             | x            | 8 bits      | 8 bits         |
| 8 bits        | 8 bits       | x           | 8 bits         |
| 16 or 24 bits | x            | 16 bits     | 16 bits        |
| 16 bits       | 8 or 16 bits | 24 bits     | 16 bits        |
| 8 or 16 bits  | 16 bits      | 24 bits     | 16 bits        |
| 24 bits       | x            | 24 bits     | 24 bits        |
| x             | 24 bits      | 24 bits     | 24 bits        |

#### Note

Whenever BS = (constant) 0, its size is considered 8 bits, and all 24 bits in B-bus are set to 0. Therefore, all operations with BS = (constant) 0 have their size determined by AS and Destination only.

CCS field (1 bit) controls whether flags will be updated or not ([Table 35-383](#)). When CCS bit exists in a microinstruction, the operation size will be used to sample flags. In some microinstructions CCS field is replaced by CCSV (2 bits, [Table 35-382](#)). Flag sampling according to CCSV can be set as defined by the operation size, or fixed as 8 or 16 bit operations.

#### Note

For the post-ALU rotate right operation, CCSV also determines the rotate size: whether 8-bit, 16-bit, or determined by the operation size.

**Table 35-382. Flag Sampling Using CCSV field**

| CCSV | Meaning                                   |
|------|-------------------------------------------|
| 00   | sample flags as an 8 bit operation        |
| 01   | sample flags as a 16 bit operation        |
| 10   | sample flags as defined by operation size |
| 11   | do not sample flags                       |

**Table 35-383. Flag Sampling Using CCS field**

| CCS | Meaning                                   |
|-----|-------------------------------------------|
| 0   | sample flags as defined by operation size |
| 1   | do not sample flags                       |

When neither CCS nor CCSV are present in the microinstruction, flags are not sampled. CCS and CCSV do not affect the Carry update on Exchange Bit operation (see [Exchange Bit](#)), but does control the N and Z flags.

### 35.15.9.2.4 B-Source Inversion

The data selected as second source (T4BBS) can be inverted (bitwise boolean NOT) before operation. This is controlled by microinstruction field BINV (1 bit, in the following table). A zero value for BINV activates B-source inversion.

**Table 35-384. B-Source Inversion - BINV**

| BINV | Meaning                      |
|------|------------------------------|
| 0    | invert B-source <sup>1</sup> |
| 1    | keep B-source bus unchanged  |

1. Except on max-constant selection, see [Generating "max" constant](#) .

BINV also selects between *adc* or *sbc* enhanced ALU operation, using inverted C flag as carry-in besides BS inversion for *sbc* . Note that BINV does not invert carry in fixed-carry operations (see [Table 35-385](#)).

When BINV = 0, T4BBS = 111 and CIN = 0, the value assigned to BS is 0x800000, instead of 0x0. See [Generating "max" constant](#) for more details.

### 35.15.9.2.5 Carry-in Control

CIN field (1 bit, [Table 35-385](#)) controls the carry-in for addition/subtraction operations. Functionality of CIN field depends on the arithmetic operation selected by ALUOP. When ALUOP is not available in microinstruction, the operation selected is *add*. For carry-in control in MDU operations, see the following table.

**Table 35-385. ALU Carry-In Control**

| operation                                         | CIN=0                            | CIN=1              |
|---------------------------------------------------|----------------------------------|--------------------|
| <i>add</i> (addition)                             | carry-in used is 1 <sup>1</sup>  | carry-in used is 0 |
| <i>adc</i> (addition with carry) <sup>2</sup>     | carry-in used is C flag          |                    |
| <i>sbc</i> (subtraction with borrow) <sup>3</sup> | carry-in used is inverted C flag |                    |

1. Except on max-constant selection, see [Generating "max" constant](#).

2. Selected by ALUOP=11000 and BINV=1

3. Selected by ALUOP=11000 and BINV=0

#### 35.15.9.2.5.1 Generating "max" constant

When T4BBS = 111, CIN = 0 and BINV = 0, BS is assigned to 0x800000 (called "max constant") instead of 0x000000. This is an exception for CIN and BINV fields: when "max constant" is selected, the carry in is 0 and B-source ("max constant" itself) is not inverted, neither the carry out.

"Max constant" is the value which, added to a time base value minus 1, gives the farthest wrapped time base value that satisfies a channel greater-equal comparison. See [Enhanced Channels](#) for more info.

### 35.15.9.2.6 Shift Operations

There are three types of shift operations: ALU, post-ALU and Shift Register. ALU shift operations are covered in [ALU/MDU Operation Selection](#). Post-ALU and Shift Register are covered in the following sections.

#### 35.15.9.2.6.1 Shift Register Operations

SR can be used as a general purpose register and it can easily shift-right its contents, combined or not to post-ALU shift operations. If field SRC (1 bit) in microcode is 0, SR will shift its contents 1 bit to the right according to the algorithmic description below. SR shifting operation depends also on SHF or ALUOP fields. ALUOP and SHF never exist both on the same microinstruction format.



**Table 35-386. Shift Register Control - SRC**

| SRC | Meaning              |
|-----|----------------------|
| 0   | shift right SR 1 bit |
| 1   | no shift             |

**SR Operation:**

```

SR[22:0] = SR[23:1];
if SHF == "01" or ALUOP == "10110" then
    SR[23] = ALU_OUT[0];
else
    SR[23] = 0;
endif;

```

**35.15.9.2.6.2 Post-ALU shift operations**

Post-ALU shift can be selected by SHF field (2 bits) or by some specific ALUOP field values. SHF and ALUOP fields are never both available in the same microinstruction format. When selecting post-ALU shift operation using ALUOP field, ALU will always add the sources before shifting the result.

**Table 35-387. Post-ALU shift operation**

| Post ALU operation   | SHF <sup>1</sup> | ALUOP                  |
|----------------------|------------------|------------------------|
| shift left (1 bit)   | 00               | 10101                  |
| shift right (1 bit)  | 01               | 10110                  |
| rotate right (1 bit) | 10               | 10111                  |
| no shift/rotate      | 11               | any other <sup>2</sup> |

1. ALU performs AS+BS before shift/rotate for all SHF values.
2. Some ALUOP combinations perform shift/rotate, but not using the Post-ALU Shifter (see [Table 1](#))

Carry flag is only updated when CCS or CCSV fields allow it (see [Flags Sampling Control](#)). Algorithmic descriptions of post-ALU shift operations are presented in [ALU ADD operation with and without shifting](#).

**35.15.9.2.7 Conditional ALU/MDU Operation Execution**

The 3-bit field AS/CE allows conditional execution of arithmetic operation, as shown in the following table. The same field can also be used for overriding the size of A-Source (see [A-Source Size Override](#)).

**Table 35-388. ALU/MDU Conditional Execution**

| AS/CE | Meaning                                                                       |
|-------|-------------------------------------------------------------------------------|
| 000   | used for A-Source size override (see <a href="#">A-Source Size Override</a> ) |

*Table continues on the next page...*

**Table 35-388. ALU/MDU Conditional Execution (continued)**

| AS/CE | Meaning                                  |
|-------|------------------------------------------|
| 001   |                                          |
| 010   | execute if C = 1                         |
| 011   | execute if C = 0                         |
| 100   | execute if Z = 1                         |
| 101   | execute if Z = 0                         |
| 110   | execute if N = 1                         |
| 111   | execute unconditionally/no size override |

Other operations not related to ALU/MDU in the same microinstruction are not affected by the AS/CE field.

If a conditional operation is selected, there is no A-Source size override; similarly, when size override for A-Source is selected, the ALU/MDU operation executes unconditionally.

When a conditional ALU/MDU operation is not executed:

- the destination register is not updated. If the destination is CHAN, no actions associated with CHAN assignment occur (see [Channel Selection Register - CHAN](#)).
- the ALU and MDU flags are not updated.
- MDU does not start any operation, i.e., MACH and MACL are not updated.
- SR does not shift.
- T4ABS-selected read-match does not occur.

### 35.15.9.2.8 A-Source Size Override

Some values of the AS/CE field are used for A-Source Size Override, as shown in the following table.

**Table 35-389. A-Source Size Override**

| AS/CE | Meaning                                                                                       |
|-------|-----------------------------------------------------------------------------------------------|
| 000   | A-source size override to 8 bits                                                              |
| 001   | A-source size override to 16 bits                                                             |
| 010   | used for conditional execution (see <a href="#">Conditional ALU/MDU Operation Execution</a> ) |
| 011   |                                                                                               |
| 100   |                                                                                               |
| 101   |                                                                                               |
|       |                                                                                               |

*Table continues on the next page...*

**Table 35-389. A-Source Size Override (continued)**

| AS/CE | Meaning                                  |
|-------|------------------------------------------|
| 110   |                                          |
| 111   | execute unconditionally/no size override |

Register size override zero-pads an overridden source to 24-bits (if no sign extension is performed, see [A-source Sign Extension](#)) and affects operation size calculation. When register source is wider than size override, most significant bits of selected register are not used as A source (zeros are used instead). When size override is wider than selected register, register value is padded to zeros.

When size override is used with MDU operations, it affects only the operand values, but not the operation size: MDU operation size is fully determined by the operation definition (fields ALUOP, ALUOPI).

**Table 35-390. AS/CE field A source size override functionality**

| Size Override | Size of Selected Register | AS Value <sup>1</sup> |
|---------------|---------------------------|-----------------------|
| 8 bits        | 8 bits                    | reg[7:0]              |
| 8-bits        | 16 bits                   | reg[7:0]              |
| 8bits         | 24 bits                   | reg[7:0]              |
| 16 bits       | 8 bits                    | reg[15:0]             |
| 16 bits       | 16 bits                   | reg[15:0]             |
| 16 bits       | 24 bits                   | reg[15:0]             |

1. All values are zero-padded to 24 bits

### 35.15.9.2.9 A-source Sign Extension

The SEXT microinstruction field forces sign extension of A source according to the size of A operand, either overridden or not by AS/CE field, according to the following table. The sign is taken from the size-overridden value, not the original one.

**Table 35-391. A source Sign Extension**

| SEXT | Meaning                                           |
|------|---------------------------------------------------|
| 0    | extends sign of A source from its size to 24-bits |
| 1    | does not extend sign of A source                  |

A-source sign is not extended in microinstructions without SEXT field, even if AS/CE field is present.

### 35.15.9.2.10 ALU/MDU Operation Selection

When field ALUOP is available in microinstruction, enhanced ALU operations shown in the following table can be performed, otherwise addition is performed. The ALU operations are defined in [ALU and Post-ALU Shifter](#). The MDU operations are defined in [MAC and Divide Unit \(MDU\)](#).

**Table 35-392. ALU Operation Selection - ALUOP**

| ALUOP | Operation                           | Comment                                                                                            |
|-------|-------------------------------------|----------------------------------------------------------------------------------------------------|
| 00000 | AS mults BS[7:0]                    | signed multiplication                                                                              |
| 00001 | AS multu BS[7:0]                    | unsigned multiplication                                                                            |
| 00010 | AS fmults BS[7:0]                   | signed fractional multiplication                                                                   |
| 00011 | AS fmultu BS[7:0]                   | unsigned fractional multiplication                                                                 |
| 00100 | AS mults BS[15:0]                   | signed multiplication                                                                              |
| 00101 | AS multu BS[15:0]                   | unsigned multiplication                                                                            |
| 00110 | AS fmults BS[15:0]                  | signed fractional multiplication                                                                   |
| 00111 | AS fmultu BS[15:0]                  | unsigned fractional multiplication                                                                 |
| 01000 | AS mults BS[23:0]                   | signed multiplication                                                                              |
| 01001 | AS multu BS[23:0]                   | unsigned multiplication                                                                            |
| 01010 | AS macs BS[23:0]                    | signed multiply-accumulate                                                                         |
| 01011 | AS macu BS[23:0]                    | unsigned multiply-accumulate                                                                       |
| 01100 | AS div BS[7:0]                      | unsigned division by 8-bit value                                                                   |
| 01101 | AS div BS[15:0]                     | unsigned division by 16-bit value                                                                  |
| 01110 | AS div BS [23:0]                    | unsigned division by 24-bit value                                                                  |
| 01111 | n.a.                                | RESERVED                                                                                           |
| 10000 | AS[23:0]   BS[23:0]                 | 24 bit bitwise OR                                                                                  |
| 10001 | AS[23:0] ^ BS[23:0]                 | 24 bit bitwise XOR                                                                                 |
| 10010 | AS[23:0] & BS[23:0]                 | 24 bit bitwise AND                                                                                 |
| 10011 | abs(AS)                             | absolute value of AS                                                                               |
| 10100 | AS + BS                             | arithmetic addition                                                                                |
| 10101 | (AS + BS) shl 1                     | arithmetic addition with 1-bit post-ALU shift left. ( <a href="#">Post-ALU shift operations</a> )  |
| 10110 | (AS + BS) shr 1                     | arithmetic addition with 1-bit post-ALU shift right ( <a href="#">Post-ALU shift operations</a> )  |
| 10111 | (AS + BS) ror 1                     | arithmetic addition with 1-bit post-ALU rotate right ( <a href="#">Post-ALU shift operations</a> ) |
| 11000 | AS adc/sbc BS <sup>1</sup>          | addition/subtraction with C flag ( <a href="#">Carry-in Control</a> )                              |
| 11001 | AS shl (2 <sup>^(BS[1:0]+1)</sup> ) | AS is shifted left: 2 bits for BS=0; 4 for BS=1; 8 for BS=2; 16 for BS=3                           |
| 11010 | AS shr (2 <sup>^(BS[1:0]+1)</sup> ) | AS is shifted right: 2 bits for BS=0; 4 for BS=1; 8 for BS=2; 16 for BS=3                          |
| 11011 | AS ror (2 <sup>^(BS[1:0]+1)</sup> ) | AS is rotated right: 2 bits for BS=0; 4 for BS=1; 8 for BS=2; 16 for BS=3                          |
| 11100 | AS exch BS[4:0]                     | exchange C flag and AS bit determined by BS ( <a href="#">Exchange Bit</a> )                       |
| 11101 | AS setb BS[4:0]                     | set bit in AS determined by BS <sup>2</sup>                                                        |
| 11110 | AS clrb BS[4:0]                     | clear bit in AS determined by BS <sup>3</sup>                                                      |
| 11111 | n.a.                                | RESERVED                                                                                           |

1. Addition/Subtraction is selected by field BINV (see [B-Source Inversion](#))

2. In setb and clrb operations, the register that drives A source is not changed, unless selected as destination of the operation.
3. In setb and clrb operations, the register that drives A source is not changed, unless selected as destination of the operation.

### 35.15.9.2.11 Operations With Immediate Data

Immediate data can be used with some specific microinstruction formats. eTPU microcode allows 8-, 16- or 24-bit immediate data. Immediate data is loaded as B-source, so T4BBS field is not available. When using 24-bit immediate, an add is performed with A-source = 0 and flags are not updated. ALU operations are available with 8-bit immediate, although the field that selects ALU operation in this case is ALUOPI.

#### 35.15.9.2.11.1 24-bit Immediate Destination

When using 24-bit immediate data, the destination register is selected by T2D field (2 bits), according to the following table.

**Table 35-393. 24-bit Immediate Destination - T2D**

| T2D | Target Register |
|-----|-----------------|
| 00  | P[23:0]         |
| 01  | A               |
| 10  | SR              |
| 11  | DIOB            |

#### 35.15.9.2.11.2 Enhanced ALU Operations With Immediate Data

Enhanced operations with immediate data, selected by ALUOPI (5 bits) are allowed only with an 8 bit immediate operand (see the following table).

**Table 35-394. ALU Operation Selection With Immediate Data - ALUOPI**

| ALUOPI | Operation                                         | Comment                            |
|--------|---------------------------------------------------|------------------------------------|
| 00000  | AS mults #imm8                                    | signed multiplication              |
| 00001  | AS multu #imm8                                    | unsigned multiplication            |
| 00010  | AS fmults #imm8                                   | signed fractional multiplication   |
| 00011  | AS fmultu #imm8                                   | unsigned fractional multiplication |
| 00100  | AS div #imm8                                      | unsigned division                  |
| 00101  | n.a.                                              | reserved                           |
| 00110  | n.a.                                              | reserved                           |
| 00111  | n.a.                                              | reserved                           |
| 01000  | AD[7:0] = AS[7:0]   #imm8, AD[23:8]<br>= AS[23:8] | bitwise OR                         |

*Table continues on the next page...*

**Table 35-394. ALU Operation Selection With Immediate Data - ALUOPI (continued)**

| ALUOPI | Operation                                                                           | Comment                                                                                              |
|--------|-------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------|
| 01001  | $AD[7:0] = AS[7:0] \wedge \#imm8$ , $AD[23:8] = AS[23:8]$                           | bitwise XOR                                                                                          |
| 01010  | $AD[7:0] = AS[7:0] \& \#imm8$ , $AD[23:8] = AS[23:8]$                               | bitwise AND                                                                                          |
| 01011  | $AD[7:0] = AS[7:0] \& \#imm8$ , $AD[23:8] = 0 \times 0$                             | bitwise AND with clear                                                                               |
| 01100  | $AD[15:8] = AS[15:8] \vee \#imm8$ , $AD[23:16] = AS[23:16]$ , $AD[7:0] = AS[7:0]$   | bitwise OR                                                                                           |
| 01101  | $AD[15:8] = AS[15:8] \wedge \#imm8$ , $AD[23:16] = AS[23:16]$ , $AD[7:0] = AS[7:0]$ | bitwise XOR                                                                                          |
| 01110  | $AD[15:8] = AS[15:8] \& \#imm8$ , $AD[23:16] = AS[23:16]$ , $AD[7:0] = AS[7:0]$     | bitwise AND                                                                                          |
| 01111  | $AD[15:8] = AS[15:8] \& \#imm8$ , $AD[23:16] = 0 \times 0$ , $AD[7:0] = 0 \times 0$ | bitwise AND with clear                                                                               |
| 10000  | $AD[23:16] = AS[23:16] \vee \#imm8$ , $AD[15:0] = AS[15:0]$                         | bitwise OR                                                                                           |
| 10001  | $AD[23:16] = AS[23:16] \wedge \#imm8$ , $AD[15:0] = AS[15:0]$                       | bitwise XOR                                                                                          |
| 10010  | $AD[23:16] = AS[23:16] \& \#imm8$ , $AD[15:0] = AS[15:0]$                           | bitwise AND                                                                                          |
| 10011  | $AD[23:16] = AS[23:16] \& \#imm8$ , $AD[15:0] = 0 \times 0$                         | bitwise AND with clear                                                                               |
| 10100  | $AS + \#imm8$                                                                       | arithmetic addition                                                                                  |
| 10101  | $(AS + \#imm8) \text{ shl } 1$                                                      | arithmetic addition with 1-bit shift left.                                                           |
| 10110  | $(AS + \#imm8) \text{ shr } 1$                                                      | arithmetic addition with 1-bit shift right                                                           |
| 10111  | $(AS + \#imm8) \text{ ror } 1$                                                      | arithmetic addition with 1-bit rotate right                                                          |
| 11000  | n.a.                                                                                | reserved                                                                                             |
| 11001  | $AS \text{ shl } (2^{(\#imm8[1:0]+1)})$                                             | AS is shifted left: 2 bits for $\#imm8=0$ ; 4 for $\#imm8=1$ ; 8 for $\#imm8=2$ ; 16 for $\#imm8=3$  |
| 11010  | $AS \text{ shr } (2^{(\#imm8[1:0]+1)})$                                             | AS is shifted right: 2 bits for $\#imm8=0$ ; 4 for $\#imm8=1$ ; 8 for $\#imm8=2$ ; 16 for $\#imm8=3$ |
| 11011  | $AS \text{ ror } (2^{(\#imm8[1:0]+1)})$                                             | AS is rotated right: 2 bits for $\#imm8=0$ ; 4 for $\#imm8=1$ ; 8 for $\#imm8=2$ ; 16 for $\#imm8=3$ |
| 11100  | $AS \text{ exch } \#imm8[4:0]$                                                      | exchange C flag and AS bit determined by $\#imm8[4:0]$ (see <a href="#">Exchange Bit</a> )           |
| 11101  | n.a.                                                                                | reserved                                                                                             |
| 11110  | n.a.                                                                                | reserved                                                                                             |
| 11111  | n.a.                                                                                | reserved                                                                                             |

### 35.15.9.3 Channel Control and Configuration Microoperations

Channel Control and Configuration fields set configuration values in the channel logic of the channel selected by the CHAN register, except fields LSR and CIRC.

#### 35.15.9.3.1 Channel flags operations

Each channel has two associated hardware flags, called Channel Flag 0 and Channel Flag 1. Microcode field FLC (3 bits) allows them to be set or cleared, as shown in the following table. These flags can be tested by microcode as a branch condition, and may also influence in the Entry Point taken, allowing fast state decoding. For more details, see [Entry Points](#).

**Table 35-395. P flags operation - FLC**

| FLC | Meaning                        |
|-----|--------------------------------|
| 000 | Clear flag0                    |
| 001 | Set flag0                      |
| 010 | Clear flag1                    |
| 011 | Set flag1                      |
| 100 | Copy flag1:flag0 from P[25:24] |
| 101 | Copy flag1:flag0 from P[27:26] |
| 110 | Copy flag1:flag0 from P[29:28] |
| 111 | No operation                   |

#### 35.15.9.3.2 Comparator and Time Base Selection

TBSA and TBSB fields (4-bit wide each) are used to configure the type of the comparator and the time bases used for match or capture (see the following tables). TBSA can also be used to control the Output Buffer Enable signal (See [Pin Control Registers](#)).

**Table 35-396. Time Base Selection A - TBSA**

| TBSA[3] = 0 | TBSA bit               | 2                    | 1                 | 0                  |
|-------------|------------------------|----------------------|-------------------|--------------------|
|             | bitfield               | Comparator selection | Capture selection | Match TB selection |
|             | 0                      | greater or equal     | TCR1              | TCR1               |
|             | 1                      | equal-only           | TCR2              | TCR2               |
| TBSA[3] = 1 | action                 | 2                    | 1                 | 0                  |
|             | set OBE=1 <sup>1</sup> | 0                    | 0                 | 0                  |

*Table continues on the next page...*

**Table 35-396. Time Base Selection A - TBSA (continued)**

|  |                        |                  |   |   |
|--|------------------------|------------------|---|---|
|  | set OBE=0 <sup>1</sup> | 0                | 0 | 1 |
|  | do nothing             | 1                | 1 | 1 |
|  | reserved               | all other values |   |   |

1. Output Buffer Enable is not available on this chip.

**Table 35-397. Time Base Selection B - TBSB**

|             | TBSB bit   | 2                    | 1                 | 0                  |
|-------------|------------|----------------------|-------------------|--------------------|
| TBSB[3] = 0 | bitfield   | Comparator selection | Capture selection | Match TB selection |
|             | 0          | greater or equal     | TCR1              | TCR1               |
|             | 1          | equal-only           | TCR2              | TCR2               |
|             |            |                      |                   |                    |
| TBSB[3] = 1 | action     | 2                    | 1                 | 0                  |
|             | do nothing | 1                    | 1                 | 1                  |
|             | reserved   | all other values     |                   |                    |

### 35.15.9.3.3 Transition Detection and Pin Action Control

IPACA/B and OPACA/B fields are used to configure transition detection sensitivity (for the channel input signal) or output pin action control (for the channel output signal), as defined in the table below. IPACA and IPACB have the same format, where IPACA is related to Match A and first transition detection, and IPACB to Match B and second transition detection. The same applies in analogue way to OPACA and OPACB.

For the output signal, configuring OPAC registers does not change the current signal state, but defines the action to be done when a match or transition detection occurs. See [Match Recognition](#) and [Channel Modes on Output Signal Generation](#) for more information.

IPACA/B=1xx also enables assertion of MRLA/B during Time Slot Transition. See [Match Recognition](#).

**Table 35-398. Input and Output Pin Action Control - IPACA/B and OPACA/B**

| value | IPAC meaning                                  | OPAC meaning                                 |
|-------|-----------------------------------------------|----------------------------------------------|
| 000   | Do not detect transitions                     | Do not change output signal                  |
| 001   | Detect rising edge only                       | Match <sup>1</sup> sets output signal high   |
| 010   | Detect falling edge only                      | Match <sup>1</sup> sets output signal low    |
| 011   | Detect both edges                             | Match <sup>1</sup> toggles output signal     |
| 100   | Detect input signal = 0 on Match <sup>1</sup> | Transition detection sets output signal low  |
| 101   | Detect input signal = 1 on Match <sup>1</sup> | Transition detection sets output signal high |

*Table continues on the next page...*



**Table 35-398. Input and Output Pin Action Control - IPACA/B and OPACA/B (continued)**

| value | IPAC meaning       | OPAC meaning                               |
|-------|--------------------|--------------------------------------------|
| 110   | reserved           | Transition detection toggles output signal |
| 111   | do not change IPAC | do not change OPAC                         |

1. Match A is used for IPACA/OPACA, and Match B for IPACB/OPACB.

### 35.15.9.3.4 Immediate Pin State Control

It is possible to change output signal state immediately by using PSC (2 bits) and PSCS (1 bit) fields.

**Table 35-399. Immediate Pin State Control - PSC and PSCS**

| PSC | PSCS | Meaning                                                                                             |
|-----|------|-----------------------------------------------------------------------------------------------------|
| 00  | 0    | set signal as specified by OPACA (see <a href="#">Transition Detection and Pin Action Control</a> ) |
| 00  | 1    | set signal as specified by OPACB (see <a href="#">Transition Detection and Pin Action Control</a> ) |
| 01  | x    | set signal high                                                                                     |
| 10  | x    | set signal low                                                                                      |
| 11  | x    | don't change signal state                                                                           |

### 35.15.9.3.5 Write Channel Match and UDCM Registers

Match registers can have their values changed using ERWA and ERWB fields (1 bit each). They also set their respective MRLE register (see [Match Recognition](#)).

ERWA can also be used to program the UDCM register (see [UDCM - User Defined Channel Mode](#)). The field CMW selects where the contents of ERTA is copied when ERWA is active (see the following table).

**Table 35-400. Write MatchA/B - ERWA/B**

| Field | CMW | Value | Action                                                                       |
|-------|-----|-------|------------------------------------------------------------------------------|
| ERWA  | 1   | 0     | write ERTA value in MatchA. Enable matches for MatchA register (MRLEA=1)     |
|       | 0   | 0     | write ERTA value into UDCM                                                   |
|       | 1   | 1     | don't change UDCM, MatchA and MRLEA                                          |
|       | 0   | 1     | reserved                                                                     |
| ERWB  | 1   | 0     | write ERTB value in MatchB. Enable matches for MatchB register (set MRLEB=1) |
|       | 1   | 1     | don't change MatchB and MRLEB                                                |
|       | 0   |       |                                                                              |
|       | 0   | 0     | reserved                                                                     |

If ERTA or ERTB is a destination of an ALU operation and, at the same time, the respective ERWA/B field is active, the new ERTA value is the one written into the MatchA/B register or the UDCM register.

### 35.15.9.3.6 Clear Transition/Match Event Registers

Flags MRLA, MRLB, TDLA and TDLB (see [ER - Event Registers](#)) indicate the state of matches and transitions detected in the selected channel, and it is possible to clear those flags using the microcode fields MRLA, MRLB (1 bit each) and TDL (1 or 2 bits, depending on the format). The flags cleared by these microcode fields are the actual channel flags, and also the ones sampled into the branch logic.

TDL can be one or two bits wide, depending on the microinstruction format (see [Microinstruction formats](#)). Two-bit TDL allows independent clearing of TDLA and/or TDLB. [Table 35-402](#) defines the two-bit TDL field.

**Table 35-401. Clear Transition/Match Event Registers - MRLA/B, TDL**

| Field       | Meaning                                         |
|-------------|-------------------------------------------------|
| MRLA        | 0 = clear MRLA event register, 1 = don't change |
| MRLB        | 0 = clear MRLB event register, 1 = don't change |
| TDL (1 bit) | 0 = clear TDLA and TDLB flags, 1 = don't change |

**Table 35-402. Independent TDLA/B clear - two-bit TDL**

| value | Meaning                   |
|-------|---------------------------|
| 0 0   | clear TDLA                |
| 0 1   | clear TDLB                |
| 1 0   | clear both TDLA and TDLB  |
| 1 1   | do not clear TDLA or TDLB |

### 35.15.9.3.7 Disable Matches

Microcode field MRLE (1 bit) allows disabling matches on channel selected by CHAN register, for both MatchA and MatchB registers, by clearing their respective MRLE bits. Matches can be enabled for each Match register using ERWA and ERWB fields (see [Write Channel Match and UDCM Registers](#)).

**Table 35-403. Disable Matches - MRLE**

| MRLE | Meaning                                 |
|------|-----------------------------------------|
| 0    | Disable matches for Match A and Match B |
| 1    | don't change match enabling             |

Some instruction formats have a two-bit MRLE field (see [Microinstruction formats](#)) which allows independent disabling of Matches 1 and 2, as shown in the following table.

**Table 35-404. Two-bit MRLE**

| MRLE | Meaning                                      |
|------|----------------------------------------------|
| 0 0  | Disable Match A (clear MRLEA)                |
| 0 1  | Disable both Matches (clear MRLEA and MRLEB) |
| 1 0  | Disable Match B (clear MRLEB)                |
| 1 1  | nop                                          |

### 35.15.9.3.8 Disable Match and Transition Service Requests

Microcode field MTD (2 bits) disables match and transition service requests for the selected channel. MTD does not disable Link Service Request and Host Service Request. MTD sets or resets register SRI (for more details see [SRI - Match/Transition Service Request Inhibit Latch](#)) and TCCEA (see [TCCEA - Transition Continuous Capture Enable](#)) .

**Table 35-405. Disable Match and Transition Service Request - MTD**

| MTD | Action on SRI                                              | Action on TCCEA                                                   |
|-----|------------------------------------------------------------|-------------------------------------------------------------------|
| 00  | SRI = 0: enable service requests for match and transition  | TCCEA = 0: disable transition captures <sup>1</sup> when TDLA = 1 |
| 01  | SRI = 1: disable service requests for match and transition |                                                                   |
| 10  | SRI = 1: disable service requests for match and transition | TCCEA = 1: enable transition captures <sup>2</sup> when TDLA = 1  |
| 11  | don't change                                               |                                                                   |

1. Disables only captures on transition events specified by IPACA.

2. Enables only captures into CaptureA register, on transition events specified by IPACA.

### 35.15.9.3.9 Predefined Channel Modes

Microcode PDCM field (4 bits) defines the channel mode (see [Channel Modes](#)).

PDCM coding is shown in the table below. Note that PDCM bit 0 selects between Single Transition (PDCM[0]=0) and Double Transition (PDCM[0]=1) predefined modes.

PDCM is also used to select the user-defined channel mode, defined by the channel register UDCM (see [UDCM - User Defined Channel Mode](#)).

**Table 35-406. Predefined Channel Modes**

| PDCM | Channel mode |
|------|--------------|
| 0000 | em_b_st      |

*Table continues on the next page...*

**Table 35-406. Predefined Channel Modes (continued)**

| PDCM        | Channel mode                     |
|-------------|----------------------------------|
| 0001        | em_b_dt                          |
| 0010        | em_nb_st                         |
| 0011        | em_nb_dt                         |
| 0100        | m2_st                            |
| 0101        | m2_dt                            |
| 0110        | bm_st                            |
| 0111        | bm_dt                            |
| 1000        | m2_o_st                          |
| 1001        | m2_o_dt                          |
| <b>1010</b> | <b>user-defined channel mode</b> |
| 1011        | reserved                         |
| 1100        | sm_st                            |
| 1101        | sm_dt                            |
| 1110        | sm_st_e                          |
| 1111        | keep current channel mode        |

### 35.15.9.3.10 Channel Interrupt and Data Transfer Requests

Microcode can issue Interrupt Requests, Data Transfer Requests and Global Exception through CIRC field. For more information see [Interrupts and data transfer requests](#).

**Table 35-407. Channel and Data Transfer Requests - CIRC**

| CIRC | Meaning                                                            |
|------|--------------------------------------------------------------------|
| 000  | Channel Interrupt Request from selected channel                    |
| 001  | Data Transfer Request from selected channel                        |
| 010  | Channel Interrupt and Data Transfer requests from selected channel |
| 011  | Channel Interrupt and Data Transfer requests from serviced channel |
| 100  | Channel Interrupt Request from service channel                     |
| 101  | Data Transfer Request from service channel                         |
| 110  | Global Exception                                                   |
| 111  | don't request interrupt                                            |

### 35.15.9.3.11 Clear Link Service Request

Microcode LSR field (1 bit) is used to clear the Link Service Request flag of the serviced channel (may not be the one selected by CHAN). The LSR branch condition is always cleared, but not the Link Service Request, if another channel link was received by the serviced channel during the executing thread. See [Channel Link](#) for more information.

**Table 35-408. Link Service Request Negation Control - LSR**

| LSR | Meaning                               |
|-----|---------------------------------------|
| 0   | clear Link Service Request (flag LSR) |
| 1   | don't change                          |

### 35.15.9.4 Flow Control Microoperations

eTPU has *jump* and *call* microoperations to change microcode flow. Besides, eTPU has *dispatch jump* and *dispatch call* that can be used to implement a jump table. In *call* (or *dispatch call*) microoperation, the return address is saved in RAR register. If nested sub-routine calls are necessary, return address values have to be saved in a stack, usually implemented with DIOB register.

Flow Control microoperations are also provided to repeat a given microinstruction, to finish the current thread execution, and to halt the microengine.

#### 35.15.9.4.1 Ending Current Thread - END

Microcode END field (1 bit) finishes current thread and allows other channels to be serviced. If END field is 0, the current instruction is completed and the thread is finished. END = 1 has no effect, and the next microinstruction is executed Any MDU operation (see [MAC and Divide Unit \(MDU\)](#)) that could be still pending when the thread is finished is left incomplete. END also releases any semaphore locked by the Engine.

#### 35.15.9.4.2 Branch Operations

Branch operations can be **jump** or **call**. The target address of *jump* or *call* microoperations is always immediate and absolute. Branch microoperation is affected by FLS field (refer to [Flush Pipeline](#)).

##### 35.15.9.4.2.1 Selecting Jump or Call Microoperations

The only difference between *jump* and *call* microoperations is that when a *call* is executed the value of PC or PC+1 (depending on flush, see [Flush Pipeline](#)) is saved in RAR register. The microcode field J/C (1 bit) selects whether *jump* or a *call* is executed, according to the following table.

**Table 35-409. Jump / Call Selection - J/C**

| J/C | Meaning |
|-----|---------|
| 0   | jump    |
| 1   | call    |

### 35.15.9.4.2.2 Branch Target Address

Microcode BAF field (14 bits) indicates the absolute address of a *jump/call target*.

### 35.15.9.4.2.3 Conditional/unconditional branch

*Jump* and *call* can be conditional or unconditional, depending on the BCC (6 bits) and BCF (1 bit) fields, as shown in [Table 35-410](#) and [Table 35-411](#). BCF determines whether branch is taken when condition specified by BCC is true or false. When a branch condition uses the channel flags, the channel context is related to the channel number written in CHAN register.

**Table 35-410. Branch Condition Inversion - BCF**

| BCF | Meaning                                        |
|-----|------------------------------------------------|
| 0   | branch if condition determined by BCC is false |
| 1   | branch if condition determined by BCC is true  |

**Table 35-411. Branch Condition Selection - BCC**

| BCC    | Meaning           | BCC    | Meaning                                                       |
|--------|-------------------|--------|---------------------------------------------------------------|
| 001110 | Flag 0            | 001111 | Flag 1                                                        |
| 100000 | V ALU flag        | 110000 | PSS channel flag                                              |
| 100001 | N ALU flag        | 110001 | PRSS channel flag                                             |
| 100010 | C ALU flag        | 110010 | "Less Than" ALU flag combination (signed) <sup>1</sup>        |
| 100011 | Z ALU flag        | 110011 | "Lower or Equal" ALU flag combination (unsigned) <sup>2</sup> |
| 100100 | MV MDU flag       | 110100 | P[24]                                                         |
| 100101 | MN MDU flag       | 110101 | P[25]                                                         |
| 100110 | MC MDU flag       | 110110 | P[26]                                                         |
| 100111 | MZ MDU flag       | 110111 | P[27]                                                         |
| 101000 | TDLA channel flag | 111000 | P[28]                                                         |
| 101001 | TDLB channel flag | 111001 | P[29]                                                         |
| 101010 | MRLA channel flag | 111010 | P[30]                                                         |
| 101011 | MRLB channel flag | 111011 | P[31]                                                         |
| 101100 | LSR channel flag  | 111100 | PSTO channel flag                                             |
| 101101 | MB flag MDU flag  | 111101 | PSTI channel flag                                             |

Table continues on the next page...

**Table 35-411. Branch Condition Selection - BCC (continued)**

| BCC                       | Meaning            | BCC    | Meaning              |
|---------------------------|--------------------|--------|----------------------|
| 101110                    | FM[1] channel flag | 111110 | SMLCK semaphore flag |
| 101111                    | FM[0] channel flag | 111111 | false                |
| all other values reserved |                    |        |                      |

1. "less than" is a signed comparison, equal to the xor between ALU flags V and N; e.g.,  $0 < 0xFFFFFFFF$  tests as false ( $0 < -1$ ).
2. "lower equal" is an unsigned comparison, equal to Z or C; e.g.,  $0 < 0xFFFFFFFF$  tests as true.

### 35.15.9.4.3 Dispatch Microoperation

Dispatch microoperation is an unconditional branch where the target address is always  $PC+P[31:24]$  (unsigned). Dispatch is affected by FLS field (refer to [Flush Pipeline](#)). Dispatch microoperation is defined by R/D field (2 bits, see the following table). Field R/D can also be used to define return from sub-routine (see [Return from subroutine](#)).

**Table 35-412. Return and Dispatch - R/D**

| R/D | Meaning                                                              |
|-----|----------------------------------------------------------------------|
| 00  | return from subroutine (see <a href="#">Return from subroutine</a> ) |
| 01  | dispatch jump                                                        |
| 10  | dispatch call                                                        |
| 11  | don't change microinstruction flow                                   |

### 35.15.9.4.4 Return from subroutine

When a subroutine *call* or a *dispatch call* microoperation is executed, the return address is saved in RAR register. To return from a subroutine a microoperation is available to load the contents of RAR register back to the PC. Fields R/D (2 bits) or RTN (1 bit, see the table below) can be used to return from subroutine. For R/D field, see [Table 35-412](#).

Return from subroutine microoperation is affected by FLS (see [Flush Pipeline](#)) when field R/D is used. Return execution through RTN always flushes the pipeline.

**Table 35-413. Return from subroutine - RTN**

| RTN | Meaning                    |
|-----|----------------------------|
| 0   | Return with pipeline flush |
| 1   | Do not return              |

### 35.15.9.4.5 Flush Pipeline

When a branch, dispatch or subroutine return microoperation is executed, the next microinstruction can be executed unconditionally before the flow change takes effect, since microengine has a three-stage pipeline. Executing the next microinstruction after a branch maximizes execution performance. This feature is controlled by field FLS (1 bit, see the table below). When FLS=0 the pipeline is flushed, so the next microinstruction placed after a branch is decoded as NOP if the branch is taken. If FLS=1, the microinstruction placed after the branch is executed, either if the branch is taken or not, as shown in the figure below.

Flush also controls which value is stored in RAR in a call: in case of no flush, it is the address of the branch/dispatch instruction + 2, even if RAR is the ALU destination of the instruction after the call; in case of a flush, it is the address of the instruction following branch/dispatch.

If a branch with no flush is followed by another branch with no flush, the instructions are executed in the following order:

1. first branch
2. second branch
3. first branch's destination instruction
4. second branch's destination instruction, and the flow proceeds normally from then on

The destination of the first branch must not be another flow changing instruction (branch, return or dispatch). Similar flows apply when returns or dispatches are used instead of branches. This scheme can be used to implement quick table look-ups with a dispatch replacing the first branch, for instance.

**Table 35-414. Flush Pipeline - FLS**

| FLS | Meaning                                                                                     |
|-----|---------------------------------------------------------------------------------------------|
| 0   | flush pipeline when jump / call / dispatch jump / dispatch call / return is executed        |
| 1   | do not flush pipeline when jump / call / dispatch jump / dispatch call / return is executed |



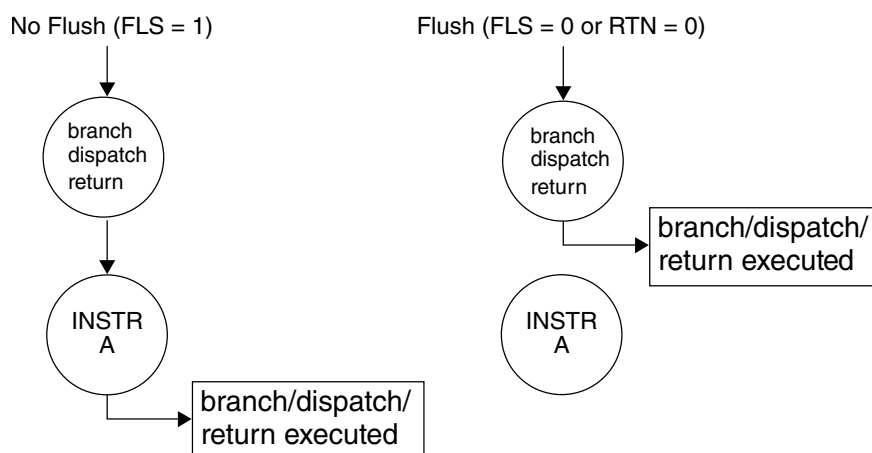


Figure 35-331. Flush Pipeline

### 35.15.9.4.6 HALT microinstruction

HALT is a microinstruction provided to implement software breakpoints (see [Software breakpoints](#)). Note that HALT is coded as a microinstruction format, not a field (see [Microinstruction formats](#)). The execution of this instruction puts the microengine in halt state. For more information about the implications of microengine halt state, see [Microengine Halt State](#). HALT is valid only if software breakpoints are enabled at the Debug interface. If software breakpoints are not enabled, HALT executes as a NOP and is treated as an Illegal Instruction (see [Illegal instructions](#)).

### 35.15.9.4.7 NOP microinstruction

There is not a unique microinstruction with an assigned opcode to do No Operation. NOP microinstruction is achieved through any of the formats shown on [Microinstruction formats](#) where the user can assign to each individual field the corresponding value for "No Operation". However, to prevent future impacts of instruction changes on object code compatibility, the instruction value 0x4FFFFFFF should always be used for NOP.

## 35.15.9.5 Illegal instructions

An instruction is considered illegal if any reserved field value is used, including when the fields marked rsv in the instruction formats (see [Microinstruction formats](#)) are assigned value 0. A HALT instruction is exceptionally considered illegal when executed with software breakpoints disabled (see [HALT microinstruction](#)). Global Exception may be issued up to two microcycles after instruction fetch. The execution results of an illegal instruction on the microengine, channel logic or host interface are unspecified, except for the HALT case.

If the microengine decodes an illegal instruction, the following actions are taken:

- A Global Exception is issued.
- The ILF1/2 flags in [eTPU Module Configuration Register \(eTPU\\_SCR\\_MCR\)](#) are set to indicate this occurrence to the Host.
- A breakpoint occurs, if NDEDI is present and configured to do so (see the NDEDI details for more information).

### **35.15.9.6 Microinstruction Parallelism Issues**

This section clarifies parallelism issues that arise when two non-commutative microoperations appear in the same microinstruction.

#### **35.15.9.6.1 ALU Operations and Read Match Registers**

ALU operations have only one destination register, but there is one case where source selection determines destination: read Match register in ERTA and ERTB registers. In this case if ALU destination is ERTA or ERTB a conflict arises. The ALU destination value overwrites the value read from the match registers.

#### **35.15.9.6.2 ALU and SDM Operations**

P and DIOB registers can be selected as destination by both ALU and SDM (read) microoperations in the same microinstruction. Since P and DIOB update from SDM data happens after P and DIOB update for ALU/MDU microoperations, the data read from SDM remains in P or DIOB after an operation when one of them is specified as destination for both ALU and SDM microoperations. In this case, the value loaded into P or DIOB is the one read from SDM. However, the ALU operation is executed and its flags are updated accordingly.

When P or DIOB is destination of an SDM read and also an ALU source at the same microinstruction, the value before the read is used for the ALU operation.

If DIOB is the ALU destination and P is loaded from SDM or vice-versa, no conflict occurs, and the result is the same as if operations occurred separately.

All the above also applies to Zero SDM operations.

When using P or DIOB as destination for ALU operations and also as source for a SDM write operation, the data written in SDM is the one calculated by ALU, which means it is possible to calculate a value and write it in an SDM address using only one microinstruction.

The old value of DIOB or the old value minus 4 (pre-decrement) is always used when DIOB is selected as address (indirect address mode), no matter if DIOB is selected as destination of either the SDM or ALU. For the value loaded into DIOB, refer to the following table.

**Table 35-415. DIOB load from SDM and ALU**

| DIOB selected as<br>SDM read destination? | DIOB selected as<br>ALU destination? | DIOB load value                                                |
|-------------------------------------------|--------------------------------------|----------------------------------------------------------------|
| no                                        | no                                   | DIOB,<br>--DIOB (pre-decrement),<br>or DIOB++ (post-increment) |
| yes                                       | no                                   | SDM read data<br>(post-inc and pre-dec ignored)                |
| yes                                       | yes                                  | SDM read data<br>(post-inc, pre-dec and ALU result ignored)    |
| no                                        | yes                                  | ALU result<br>(post-inc and pre-dec ignored)                   |

When CHAN is the destination of an ALU operation in parallel with a channel-relative SDM access, the CHAN value used to calculate the address is the one before the ALU operation.

### 35.15.9.6.3 ERTA/B as ALU destination and ERWA/B

The value in ERTA and ERTB registers can be written in match registers of the selected channel by using fields ERWA and ERWB ([Write Channel Match and UDCM Registers](#)). If, at the same microinstruction, ERTA or ERTB is destination of an ALU/MDU microoperation, the value written in Match registers is the ALU/MDU result.

The same applies to UDCM when ERTA is the destination of an ALU operation and instruction fields ERWA and CMW are active .

If an ALU operation occurs in parallel with ERWA/B but ERTA/B are not the destination of an ALU/MDU operation, then UDCM and MatchA/B receives the ERTA/B value.

#### 35.15.9.6.4 ERWA/B and MRLE

ERWA/B automatically sets the MRLEA/B channel latch, respectively (see [Write Channel Match and UDCM Registers](#)). Microinstruction fields ERWA/B independently sets MRLEA/B channel flags, regardless of MRLE.

#### 35.15.9.6.5 CHAN assignment, Read Match and ERWA/B

When CHAN is a destination of an ALU operation it causes a read of the CaptureA/B register values into ERTA/B. The Capture registers loaded into ERTA/B are selected by the new CHAN value. The value of the CaptureA/B registers overwrites any read-match commanded simultaneously.

If CHAN assignment happens with an ERWA/B operation in the same instruction, the updated Match register(s) belong to the new selected channel.

#### 35.15.9.6.6 Read Match and ERWA/B

If a read match operation is executed with ERWA/B in the same microinstruction, the MatchA/B registers receive the old values of ERTA/B, and the ERTA/B registers receive the old MatchA/B values simultaneously, i.e.: ERTA/B and MatchA/B swap their values.

If ERTA/B is the destination of an ALU operation at the same instruction, MatchA/B gets the ALU result (see [ERTA/B as ALU destination and ERWA/B](#)), but the ERTA/B not being written still receives the old MatchA/B values.

#### Note

Read match, ERWA/B and CHAN assignment can be active at the same instruction. Combining rules [CHAN assignment, Read Match and ERWA/B](#) and [Read Match and ERWA/B](#), the result is: ERTA/B receives the CaptureA/B values of the new CHAN value, and MatchA/B of the new channel receives the old ERTA/B value(s).

#### 35.15.9.6.7 Stack accesses and ALU operations

Post-increment is ignored in a stack operation (field STC) if DIOB is loaded from SDM: DIOB keeps the value read from SDM. Pre-decrement is ignored in a stack operation (field STC) if DIOB is destination of an ALU operation, for DIOB load value, but not for DIOB as address. Post-increment/pre-decrement remains valid in all other situations. These rules can be summarized in the following equivalent C code, and in [Table 35-415](#).

```
DIOB = *DIOB[15:2];    // read without posinc/predec
*DIOB[15:2] = DIOB;    // write without posinc/predec
```

```

DIOB = * (--DIOB[15:2]); // read with predec
DIOB = *DIOB[15:2];      // read with posinc (ignored)
* (--DIOB[15:2]) = DIOB; // write with predec
*DIOB[15:2]++ = DIOB;    // write with posinc (value written is before
increment)

```

### 35.15.9.6.8 SRC and ALU/MDU operations

If operation SRC is active (field SRC = 0) and register SR is selected as destination of an ALU operation, the value of the ALU operation prevails over the shifted value.

The value of SR used as source in the ALU/MDU operation is the one before the shift.

### 35.15.9.6.9 Semaphore lock/free and SMLCK branch condition

When the SMLCK branch condition is tested at the same microinstruction of a semaphore lock or free, the condition is evaluated after the semaphore action (either free or lock) is taken.

### 35.15.9.6.10 Dispatch and SDM Read

When the most significant byte of P is read from SDM (read 8 msb bits or 32 bits) and a dispatch instruction is executed simultaneously, the dispatch target address is calculated upon the P value before the read.

### 35.15.9.6.11 CHAN Assignment, PSC/PSCS, and clear MRLEA/B, MRLA/B, TDLA/B

When clear MRLEs, MRLA/B or TDLs is done and a CHAN assignment is done at the same time, the flag selected by the old CHAN value is cleared in the channel, but the branch conditions receive the state of the flags selected by the new CHAN.

When a pin action is commanded through PCS/PSCS and a CHAN assignment is done simultaneously, the output signal affected is selected with the old CHAN value.

## 35.15.9.7 Microinstruction formats

| format | 31    | 30 | 29 | 28         | 27   | 26       | 25   | 24   | 23    | 22   | 21  | 20  | 19         | 18   | 17  | 16   | 15  | 14  | 13   | 12  | 11  | 10   | 9       | 8          | 7         | 6   | 5   | 4         | 3   | 2    | 1            | 0                        |        |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |
|--------|-------|----|----|------------|------|----------|------|------|-------|------|-----|-----|------------|------|-----|------|-----|-----|------|-----|-----|------|---------|------------|-----------|-----|-----|-----------|-----|------|--------------|--------------------------|--------|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|------|-----|-----|
| A1     | 0 0 0 |    |    | IMM[15:13] |      | IMM[7:2] |      |      |       |      |     |     | IMM[23:16] |      |     |      |     |     |      |     |     |      | IMM[12] | RTN<br>CCS | IMM[11:9] |     |     | [0:1]IMM1 |     |      | T2D          |                          | IMM[8] |      | 0 0 |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |
| A2     |       |    |    |            |      |          |      |      |       |      |     |     |            |      |     |      |     |     |      |     |     |      |         |            |           |     |     |           |     |      |              |                          |        |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |
| A3     |       |    |    |            |      |          |      |      |       |      |     |     |            |      |     |      |     |     |      |     |     |      |         |            |           |     |     |           |     |      |              |                          |        |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |
| A4     |       |    |    |            |      |          |      |      |       |      |     |     |            |      |     |      |     |     |      |     |     |      |         |            |           |     |     |           |     |      |              |                          |        |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |
| B1     | 1     | 0  | 0  | END        | CCSV | SHF      | CIN  | BINV | T4BBS |      |     | RW  | T4ABS      |      |     |      |     |     |      |     |     |      | T2ABD   |            |           |     | SRC | P/D       | CCS | rsv  | FLC<br>[1:0] | AID[7:0] (global param)  |        |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |
| B2     | 1     |    |    |            |      |          |      |      |       |      |     |     |            |      |     |      |     |     |      |     |     |      |         |            |           |     |     |           |     |      |              |                          |        |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |
| B3     | 0 0 0 |    |    |            |      |          |      |      |       |      |     |     |            |      |     |      |     |     |      |     |     |      |         |            |           |     |     |           |     |      |              |                          |        |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |
| B4     | 0     | 0  | 1  |            |      |          |      |      |       |      |     |     |            |      |     |      |     |     |      |     |     |      |         |            |           |     |     |           |     |      |              |                          |        |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |
| B5     | 0 0 1 |    |    | 0          | CCSV | SHF      | CIN  | BINV | T4BBS | RW   | FL  | rsv | TDL        | TBSA |     |      |     |     |      |     |     |      |         | TBSB       |           |     |     | LSR       | P/D | RSIZ | ZRO          | AID[6:0] (channel param) |        |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |
| B6     |       |    |    | 1          |      |          |      |      |       |      |     |     |            |      |     |      |     |     |      |     |     |      |         |            |           |     |     |           |     |      |              |                          |        |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |      |     |     |
| B7     | 0     | 1  | 1  | END        | SHF  | END      | CCSV | SHF  | END   | CCSV | SHF | END | CCSV       | SHF  | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF     | END        | CCSV      | SHF | END | CCSV      | SHF | END  | CCSV         | SHF                      | END    | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END | CCSV | SHF | END |

|                |                                   |
|----------------|-----------------------------------|
| ALU Operations | Channel Control/Config Operations |
| RAM Operations | Flow Control Operations           |

NOTES:

<sup>1</sup> format E4 is deprecated; the same result can be obtained with format E1 with field STC=11

**Figure 35-332. Microinstruction formats**

All bits marked "rsvd" are reserved, and must be coded as 1.

### 35.15.10 Shared Subsystems support

Shared Subsystems support allows eTPU to control other modules, either independently or sharing control with the host. A notable example of shared subsystem is the Reaction Module (see the Reaction Module chapter for details), which is connected to the eTPU as shown in the figure below. The host CPU accesses the Reaction Module through the eTPU, which can also access the same registers.

When the SSMAE field in [eTPU Channel n Configuration Register for Engine 1 \(eTPU\\_CCCR\\_CnCR\\_ENG1\)](#) or [eTPU Channel n Configuration Register for Engine 2 \(eTPU\\_CCCR\\_CnCR\\_ENG2\)](#) is 1, the eTPU channel's threads can control shared subsystems by accessing their registers, which are mapped onto the higher 16 Kbytes of the microcode data address space (byte addresses/DIOB values 0x4000 to 0x7FFF, equivalent to parameters 0x1000 to 0x1FFF), as shown in [Figure 35-334](#). The shared subsystem registers are accessible by microcode only with the indirect address mode (see [SDM Addressing Modes](#)).

When the SSMAE field in [eTPU Channel n Configuration Register for Engine 1 \(eTPU\\_CCCR\\_CnCR\\_ENG1\)](#) or [eTPU Channel n Configuration Register for Engine 2 \(eTPU\\_CCCR\\_CnCR\\_ENG2\)](#) is 0, the channel's threads can access only the SDM, mirrored throughout the whole DIOB address range.

The same Shared Subsystem register address space is accessible by the host through the Shared Subsystem Area within the eTPU memory map (see [Figure 35-334](#) and [Memory map](#)), allowing shared control and/or supervision between the host and the eTPU. The arbitration mechanism between host and eTPU accesses is the same used for SDM accesses (see [SDM arbitration](#)). The CDC (see [Coherent Dual-parameter Controller \(CDC\)](#)) cannot be used for coherent transfers between SDM and the Shared Subsystem Area, or within the Shared Subsystem Area itself.

#### Note

The detailed register map within the Shared Subsystem Area depends on the subsystem module.

Bus errors coming from the Shared Subsystem cause a Global Exception and are flagged by the MCR[SSAERR] bit on microengine accesses only. In the case of host access, the bus error only propagates to the slave bus, which also issues a bus error.

A Shared Subsystem may be divided into submodules or channels, hereafter called **Shared Subsystem Units** (or **Shared Units** for short), which can be individually controlled by an eTPU function (see [Functions and threads](#))

A Shared Unit is logically linked with an eTPU channel through the eTPU\_SSIR\_ETPUSSSIR<sub>n</sub> configuration<sup>38</sup>. There is one eTPU\_SSIR\_ETPUSSSIR register for each Shared Unit to configure its interface with the eTPU (see [eTPU Shared](#)

[Subsystem n Integration Register \(eTPU\\_SSIR\\_ETPUSSSIR<sub>n</sub>\)](#)). The Shared Unit uses the Link Service Request (LSR) mechanism to request service to the eTPU. The Link Service Request is targeted to the channel/engine configured in [LENG](#) and [LCHAN](#). The function assigned to this channel must be written to service the Shared Unit. The Shared Unit link service request may have the same source as an interrupt to the host.

### Note

Shared Unit link service requests are level-sensitive to the eTPU. If it comes from an interrupt source, the link service request stays asserted as long as the interrupt is asserted and enabled, so the interrupt request must be cleared before the clear of the LSR flag in the link service request thread.

[LENG](#) can also be used by the host to disable Link Service Requests to the eTPU. The eTPU microcode can only disable LSRs from the Shared Unit through configuration specific to the Shared Unit itself.

Note that more than one Shared Unit can generate LSRs to the same eTPU channel, in which case the LSR source can be determined by microcode through the Shared Unit status register(s). An eTPU channel configured as a Shared Unit LSR target can also receive Link Service Requests from other eTPU channels or Shared Units. The same LSR service and flag behavior found in links between eTPU channels applies to Shared Unit links (see [Channel Link](#)).

The table below summarizes the mapping of Reaction Module resources into Shared Subsystem Support.

---

38. However, in the case of Reaction Module, the eTPU\_SSIR\_ETPUSSSIR<sub>n</sub> configuration does not determine which eTPU channel output enables the Reaction Channel modulation; this is defined in the Reaction Module configuration.



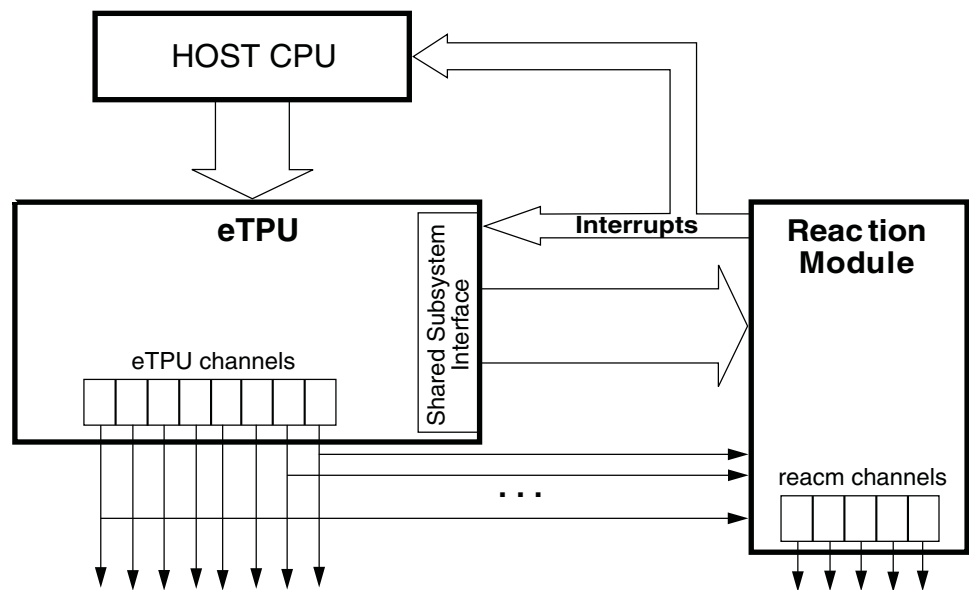


Figure 35-333. Shared Subsystem block diagram example: Reaction Module

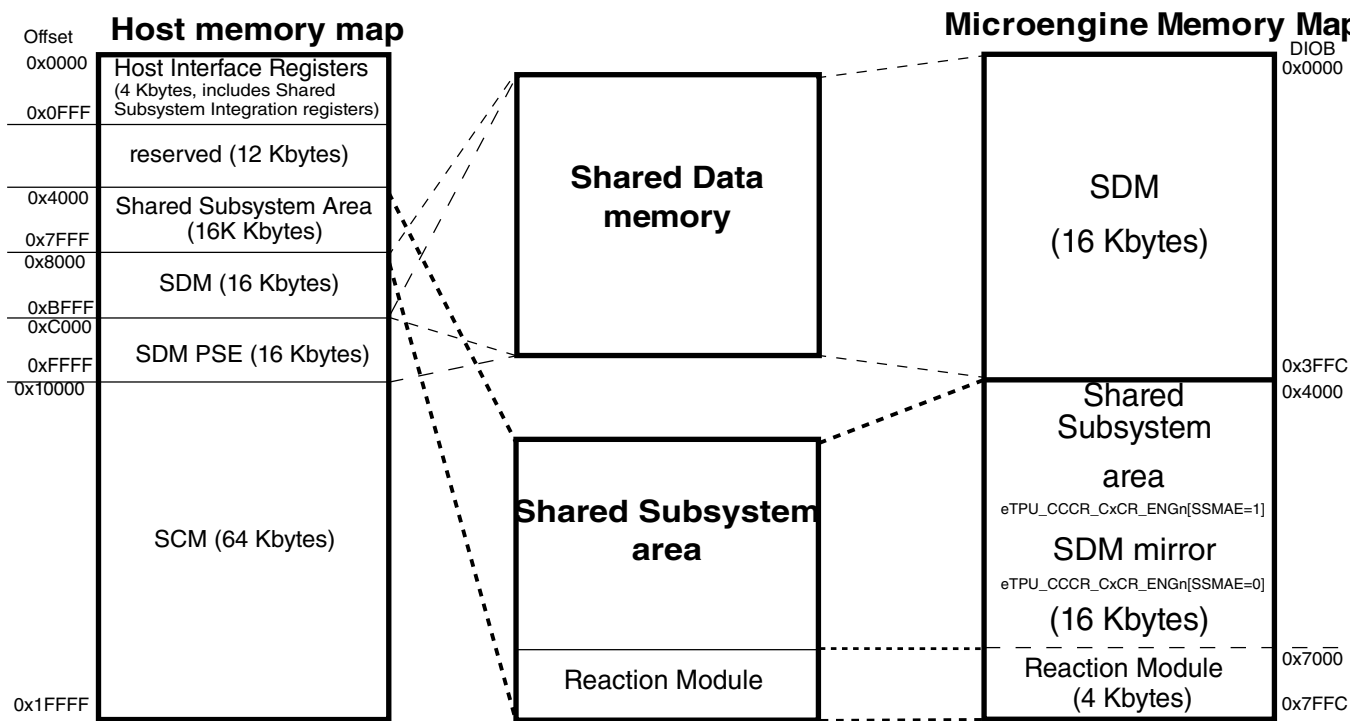


Figure 35-334. Shared Subsystem memory mapping

Table 35-416. Reaction Module Shared Subsystem support

| Reaction Module resource | Host mapping           | eTPU/microengine mapping                                                |
|--------------------------|------------------------|-------------------------------------------------------------------------|
| Register access          | eTPU Base + 0x7000     | DIOB address 0x7000<br>(indirect address mode)                          |
| Interrupts               | See Interrupts chapter | Link Service Requests mapped through<br>Subsystem Integration registers |

Table continues on the next page...

**Table 35-416. Reaction Module Shared Subsystem support (continued)**

| Reaction Module resource | Host mapping     | eTPU/microengine mapping |
|--------------------------|------------------|--------------------------|
|                          |                  | eTPU_SSIR_ETPUSSSIR $n$  |
| DMA Requests             | See eDMA chapter | Not applicable           |

### 35.15.11 Development support features

Following sections describe several features available to support development. Most debug features are accessible through a separate debug bus, and are not available through registers in the standard eTPU memory map. The details of the access to this interface are MCU-dependent (see Debug Interface chapter). The microengines can also be put in halt state by a device-level debug request.

#### NOTE

The behavior of the debug support logic is not specified in the event of a non-correctable memory error on either SCM or SDM (see [Memory error support](#)).

#### 35.15.11.1 Nexus Class 3 Support

NDEDI is an IP block designed to support Nexus functionality for the eTPU. It provides a Nexus class 3 or class 1 debug interface, depending on the MCU (see Configuration section). Nexus is a development support external interface defined by the **IEEE standard ISTO 5001-1999**. More details about it can be found in the NDEDI details.

Some of the next subsections describe debug features provided by the eTPU microengine debug support combined with the NDEDI block.

#### 35.15.11.2 Microengine Halt State

Halt is a microengine state where it suspends execution during a thread, or does not start executing a scheduled thread from idle state. While Idle State is entered from END execution without any other scheduled thread, microengine enters Halt State by any of the following events:

- execution of the HALT microinstruction (software breakpoint).

- external halt request through the Debug Interface (includes Nexus breakpoint request via EVTI input pin - see [Nexus Class 3 Support](#)).
- the other Engine enters halt state and they are configured to halt simultaneously (bit HTWIN is asserted via Nexus Interface).
- System level device debug request assertion and NDEDI register NDEDIETPUx\_DC field CBI=1. If same register's field CBT=1, microengine halts at the next time-slot boundary, if CBT=0 it halts immediately. As a particular case, microengines come halted out of reset if device debug request is asserted, since CBI reset value is 1. Microengine does not execute out of reset, either in halt (device debug request asserted) or idle state (device debug request negated), but halt enables several other features (see below).
- occurrence of any of the hardware breakpoint conditions. See [Hardware breakpoints](#) for details.
- execution of a single-step microinstruction: microengine returns to halt state after executing a single microinstruction while in halt state. See [Single-step Execution](#) and [Forced microinstruction execution](#) for details.

When microengine enters halt state, it automatically triggers the following actions:

- suspends input signal sampling and filters (respective Engine channels only), if NDEDI\_ENGINEx\_DC[PINS] = 1 at the Debug Interface.
- releases the SDM arbitration for Host or CDC accesses, no matter if microengine was halted in the middle of a dual parameter (back-to-back) access.
- stops TCR1/2 clocks of the respective Engine, if signal ndedi\_stop\_tcr is asserted at the Debug Interface.
- if the other Engine is also in halt state or stopped, allows turning the [eTPU Module Configuration Register \(eTPU\\_SCR\\_MCR\)](#) VIS bit to 1.

If all halt conditions are cleared when VIS=1, microengine(s) keep on halt state until VIS=0, when it automatically exits halt state, except on single-step (see [Single-step Execution](#)), so that single-step execution is ignored while VIS=1.

MDU continues executing until it finishes any ongoing operation even if microengine is in halt state, except when the halted instruction is an END.

There are two kinds of halt state, depending on the previous microengine state when halted:

1. **halt\_idle**, if the engine was not executing a thread when halted; the engine cannot leave **halt\_idle** to fetch instructions, so one cannot single-step or follow a program flow; it can, however, execute forced instructions (see [Forced microinstruction execution](#)).
2. **halt\_exec**, if the engine was executing a thread when halted. The engine can single-step and continue a program flow from **halt\_exec**.

When microengine exits halt state, any dependable action is suspended and, if exiting **halt\_exec**, the instruction pointed by the PC is fetched, while the instruction already fetched before halt is executed. Note that both the PC and the prefetched instructions can be modified during halt state, with a forced execution of a branch instruction (see [Forced microinstruction execution](#)).

### 35.15.11.3 Hardware breakpoints

Microengine can enter halt state through a command from the Debug Interface, configuring a hardware breakpoint. Hardware breakpoints can halt the microengine on specific conditions, listed below. These conditions depend on NDEDI configuration (see the NDEDI details for more information).

- CHAN register assignment (only by microcode, not by time slot transition).
- SDM read and/or write to a given address and/or write data. The breakpoint is always qualified by the SDM address, but the following variations are allowed:
  - break on write only, read only, or read-and-write.
  - break on higher-byte write data value, lower 24-bit write value, full word (32-bit) write value, or regardless of data. Break on read data is not supported.
- PC (program counter) value.
- Beginning of a thread with a Host Service Request pending.
- Beginning of a thread with a Link Service Request pending.
- Beginning of a thread with a Match Service Request pending.
- Beginning of a thread with a Transition Service Request pending.
- End of a thread.
- Illegal instruction execution.

All these conditions can also be qualified by the value of the CHAN register.

## NOTE

Data breakpoints are not supported on Shared Subsystem Area accesses (see [Shared Subsystems support](#)).

Hardware breakpoints halt the microengine before the halted instruction is executed, except on data breakpoints. Data breakpoints allow the instruction that performs the data access to be executed before the microengine is halted.

## Note

Data breakpoints in eTPU versions before eTPU2 halt the breakpoint instruction before its execution in all cases, including data breakpoints.

On any of these conditions, halt of one microengine does not depend on the halt of the other, unless the other engine is configured to do so, via Nexus Interface. Occurrence of any of these conditions halts the microengine, i.e., the conditions are logically ORed together, and they can be individually enabled.

While in halt state, the microengine can also execute any forced microinstruction not in the normal program flow (see [Forced microinstruction execution](#)) or, if in halt\_exec, in single-step (see [Single-step Execution](#)).

There are situations when requests for stopping an engine, breakpoint and service can occur simultaneously. Breakpoint requests always prevails over a stop request ([eTPU Engine 1 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG1\)](#) or [eTPU Engine 2 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG2\)](#) bit MDIS=1 or device debug request=1). When the eTPU is idle: stop request prevails over Service Request if there is not a hardware breakpoint request; a hardware breakpoint request leads to debug mode immediately if there is no Service Request, and after TST if there is Service Request (regardless of stop requests). The rules above are summarized in the the following table, showing the destination state of the microengine in each situation.

**Table 35-417. Breakpoint, stop and service requests resolution from idle**

| Breakpoint request | MDIS | Service request | Final state      |
|--------------------|------|-----------------|------------------|
| no                 | 0    | 0               | Idle             |
| no                 | 1    | 0               | Stop             |
| no                 | 0    | 1               | TST              |
| no                 | 1    | 1               | Stop             |
| Yes                | 0    | 0               | Halt_idle        |
| Yes                | 1    | 0               | Halt_idle        |
| Yes                | 0    | 1               | TST <sup>1</sup> |
| Yes                | 1    | 1               | TST <sup>1</sup> |

1. Breaks after TST, if signal `ndedi_sync_break` is still asserted.

When a thread is ending, it goes to Idle or TST only if there is neither a hardware breakpoint request nor a request to stop ([eTPU Engine 1 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG1\)](#) or [eTPU Engine 2 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG2\)](#) bit MDIS=1 or device debug request=1). When thread is ending and there are simultaneous hardware breakpoint and stop (MDIS=1 or device debug request=1) requests, hardware breakpoint prevails and the engine enters Debug mode (Halt\_idle state).

If the engine entered Debug mode after a thread finished (Halt\_idle state) and a "go" command comes from the debug interface, the engine state machine goes to Idle and the rules above apply. It means that if a "go" is issued in Halt\_idle state with MDIS=1, the engine goes to Idle for one microcycle and then stops (if MDIS or device debug request keeps asserted and there is no other breakpoint request).

### **Note**

Hardware breakpoint requests are ignored for the first microinstruction executed when microengine leaves halt.

## **35.15.11.4 Hardware Watchpoints**

Debug Interface allows watchpoints on the same conditions available for hardware breakpoints (see [Hardware breakpoints](#)).

### **Note**

Data breakpoints are not supported on Shared Subsystem Area accesses (see [Shared Subsystems support](#)).

## **35.15.11.5 Software breakpoints**

A software breakpoint occurs when microengine executes a HALT microinstruction. Any number of software breakpoints can be set in code, usually replacing an active microinstruction.

Like any other microinstruction, HALT increments the PC and pre-fetches the next instruction. So, before the halt state is suspended, if the original program flow must be followed, the original instruction at the HALT address must be executed, regardless if the software breakpoint is removed (replacing HALT by the original microinstruction) or not. Follows the procedure to resume execution from a software breakpoint:

1. restore the original instruction in SCM (replace HALT).
2. force a jump with flush to the original instruction (see [Forced microinstruction execution](#)).
3. if the software breakpoint must be kept: single-step and replace the original instruction with a HALT.
4. let the flow continue, issuing a GO command (leaving halt state).

Special care must be taken if HALT is followed by another HALT, and the second HALT is removed when microengine was halted by the first one. In this case, replacing the second HALT with the original microinstruction is not enough to remove the second breakpoint, because the second HALT was already prefetched and would be executed anyway when halt was suspended. The debugger must also do a forced execution of unconditional branch with flush to the original microinstruction address. That will clear the pipeline, replacing the prefetched instruction with a NOP, and load PC with the address of the removed breakpoint. So, when halt state is suspended, the original microinstruction will be fetched while NOP is executed, and program flow continues normally from then on.

### Note

A HALT instruction placed after a no-flushing branch, dispatch or return may be a problem from the debugger application standpoint: after the HALT is executed, the eTPU debug interface informs the address of the branch/dispatch/return destination, and the debugger application has no direct way to identify which HALT instruction was executed, if multiple HALTs lead to the same address. This can be solved if the debug support block (NDEDI) has a register holding the address of the last instruction executed, otherwise one should forbid non-flushed HALT instructions. See the NDEDI details.

Software breakpoint setting and removal is possible only with SCM RAM implementations or ROM implementations with SCM RAM emulation (see [SCM Emulation](#)). There is only one way of inserting software breakpoints into SCM RAM: programming VIS=1 in [eTPU Module Configuration Register \(eTPU\\_SCR\\_MCR\)](#), and then accessing SCM as an ordinary RAM from the slave bus. This can be done only if both Engines are halted or stopped.



### 35.15.11.6 Single-step Execution

When microengine is already in halt\_exec state, it can run the next microinstruction in the normal program flow and get back to halt state. PC is incremented, or assigned the BAF value in a branch with satisfied condition. Note that the executed instruction was already prefetched in the instruction pipeline, and a new microinstruction is fetched during its execution. The prefetched instruction may be cleared during halt state by the forced execution of a branch with flush (see [Forced microinstruction execution](#)), making single-step execute a NOP instead of the next instruction in the program flow.

Single-step execution is controlled by the debug interface, and is a feature available from Nexus if eTPU is connected to the NDEDI block (see NDEDI details).

The single-step execution of a NOP instruction can be useful to control input signal sampling and filtering, if NDEDI\_ENGINEx\_DC[PINS] = 1 at the Debug Interface (see NDEDI details). Single-step does not happen if VIS=1.

### 35.15.11.7 Forced microinstruction execution

When microengine is already in halt state (either halt\_idle or halt\_exec), it can run forced microinstructions through the debug interface. This feature is available from Nexus if eTPU is connected to the NDEDI block (see NDEDI details). The microinstruction, specified by the user, is not fetched from SCM and comes directly from the debug interface. MDU start commands issued by forced instructions are executed, and the MDU runs the operation until the end, independently of the halt state. The microinstruction field END is ignored.

During forced execution of any instruction except Branches, Returns and Dispatches, the PC does not change, and the prefetched instruction in the pipeline is bypassed, but not discarded. When halt state is suspended, the prefetched instruction is executed and the instruction pointed by the PC is prefetched in parallel (two-stage pipeline).

Forced execution of a Branch, Dispatch or Return loads the PC with the BAF field (if branch condition is satisfied), PC+P or RAR, respectively. If branch condition is not satisfied, PC value stays unaltered. The flush control (field FLS) also works, so that a successful forced branch with flush replaces the prefetched instruction with a NOP. So, to clear the instruction pipeline during halt, all one has to do is an unconditional branch to the desired address with flush. HALT instructions must not be executed as forced.

Forced operations that depend on the serviced channel are unspecified when executed in halt\_idle and must be avoided.



## Note

Forced instructions must not be commanded when [eTPU Module Configuration Register \(eTPU\\_SCR\\_MCR\)](#) field VIS=1.

### 35.15.11.8 Microengine Register Access

eTPU provides no direct access to microengine and channel registers from the slave bus or any other interface. However, these registers can be read and written in halt state by executing forced microinstructions (see [Forced microinstruction execution](#)). Immediate data microinstructions may be used to set register values. Some registers are not selectable for immediate data destination, so intermediary register(s) - notably P - may have to be used to carry the desired new value to the target register in two or more microinstructions. Usually the previous values of intermediary register(s) must be previously saved and restored after the whole operation.

Similar procedures apply for register reads: their contents must be dumped to SDM, where they can be read from the slave bus.

### 35.15.11.9 Microengine Flag Access

Microengine halt state allows reading the branch conditions flags through forced microinstructions or, more easily, through the NDEDI register NDEDI\_ENGINEx\_CFSR (see the NDEDI details). Flag conditions set by the user are seen by microengine for the next microinstruction execution. The flag set options are limited by the possibilities of forced microinstruction execution.

If the eTPU runs (not single-stepping) after exiting the halted state, the conditions modified during halt may remain only for the first microcycle after the halted state. After the first microcycle, branch conditions are altered only according to their regular update scheme.

### 35.15.11.10 Microengine Stall

Microengine can get into a stall state, attending a request from a debug interface signal assertion. The reason for a Stall request from NDEDI (or from any other debug support block) should be a temporary lack of resources, for instance queue full. During stall the microengine suspends execution, but all the other Engine logic continues operating: time

bases, angle logic, channel logic, input sampling and filters. Stall differs from Halt, not enabling any of the debug features that Halt enables (see [Microengine Halt State](#)). It also does not break an atomic microengine access, unlike halt.

The Microengine can be stalled when idle and from the moment TST ends, before executing the first thread microinstruction, until just before the last thread microinstruction is executed. Stall requests are ignored in any other occasions.

Microengines in a dual engine system can be independently stalled. If a forced end is issued when microengine is in stall coming from execution, the END is executed only when the microengine resumes execution from stall.

### 35.15.12 SCM Emulation

If SCM is implemented as ROM, an external RAM may be used to replace it, allowing code patching and software breakpoint setting for debugging purposes. SCM ROM replacement by Emulation RAM is MCU-dependent. The SCM may even be divided into a ROM part and a RAM part. In this case, both microengines can run code from both ROM and Emulation RAM. It is possible to make one Engine run code from RAM and the other from ROM, by using different Entry Tables. The SCM visibility conditions also apply to Emulation RAM.

All SCM implementations, either RAM, ROM or Emulation RAM, are external to the eTPU block. eTPU provides a signal to enable the switching between external SCM banks. The conditions for this switching are:

1. Both Engines stopped.
2. VIS bit = 0.

Note that these conditions also stop the clocks of the SCM interface and MISC logic.

### 35.15.13 Safety features

This section describes embedded safety features: MISC and memory error support.

The Multiple Input Signature Calculator - MISC - is an SCM test feature accessible through [eTPU Module Configuration Register \(eTPU\\_SCR\\_MCR\)](#) and [eTPU MISC Compare Register \(eTPU\\_SCR\\_MISCCMPR\)](#). MISC allows SCM test "on the fly", i.e., while eTPU is running, with no impact on eTPU functionality or performance.

Memory Error support features comprises SCM and SDM error detection, correction, report, and soft error.

### 35.15.13.1 SCM Test - Multiple Input Signature Calculator

The Multiple Input Signature Calculator (MISC) comprises special hardware that sequentially reads all SCM positions and calculates, in parallel, a 32-bit signature from a 32-input CRC signature calculator with the following polynomial:

$$1 + x^1 + x^2 + x^{22} + x^{31}$$

A complete description of the signature calculation procedure can be found in [MISC Algorithm](#).

Once started by the Host the MISC runs continuously, restarting after the completion of each cycle, when it sets [SCMMISC](#). The average time for a MISC calculation can be measured by checking SCMMISC state at regular intervals, incrementing a counter and clearing SCMMISC if it is set.

MISC accesses to the SCM array are executed if none of the engines is accessing the SCM, to avoid degradation of the microengine performance: it happens while no channel is being serviced. An ongoing MISC operation can be aborted by writing 0 to SCMMISEN.

The Host must load [eTPU MISC Compare Register \(eTPU\\_SCR\\_MISCCMPR\)](#) with the expected value to be found at the end of the MISC cycle, and then start the signature calculation by programming field SCMMISEN=1 in [eTPU Module Configuration Register \(eTPU\\_SCR\\_MCR\)](#). MISC zeroes the signature accumulator and starts reading SCM data and calculating the signature. After last SCM position is read, MISC compares the value in signature accumulator against the value in eTPU\_SCR\_MISCCMPR: if there is a mismatch MISC stops, a Global Exception is issued and [SCMMISF](#) assumes value 1. If no mismatch is found, MISC repeats the procedure automatically. When signature is being calculated, SCM address starts at the last SCM address and counts down to 0. The conditions for executing a MISC operation are (see also [Table 35-335](#)):

- Both microengines in idle state (no channel is being serviced) or stopped, in any combination (e.g., engine 1 idle with engine 2 stopped).
- [VIS](#) =0.
- [SCMMISEN](#) =1.

Note that MISC can run regardless of SCM implementation type (RAM or ROM).

If SCMMISEN=0 or VIS=1, the MISC logic stays at its initial state, with address counter pointing to the last SCM position and accumulator reset.

MISC can also be used to perform single-bit error fixes in the SCM. For more details, see [Error fixing](#).

### 35.15.13.2 Memory error support

Error Correction Code (ECC) support for SDM and SCM provides guaranteed single bit error correction and double bit error detection. Error detection and correction is performed on read accesses and on write accesses less than 32-bit wide to the SCM.

SDM ECC is calculated per byte, 5 bits wide each, totalling 20 ECC bits for each 32-bit word. That allows the ECC to be updated without a read-modify-write operation on writes of any size.

Error is not checked on bytes not selected on SDM read accesses, either by the Host, CDC or microengine. For instance, the most significant byte is not checked for errors on a microengine 24-bit read access.

Error detection also works for host accesses to the PSE area (see [Parameter Sign Extension area](#)). In this case, error is not detected in the most significant byte (bits 31:24). Also, because of sign extension, error is always detected on the bits 23:16, regardless if they are selected, whenever the most significant byte is read.

SCM ECC is 7 bits wide, comprising the whole 32-bit word. Read-modify-write is automatically made when 16-bit or 8-bit wide writes are made into the SCM in a host access. SCM errors are also detected on Entry Table accesses during TST (see [Time Slot Transition](#)) and on MISC accesses, which are considered microengine accesses with regards to interrupt generation and error reporting.

#### 35.15.13.2.1 Error detection disabling

The error detection logic in SDM and SCM can be disabled by setting the [eTPU Memory Error Control Register \(eTPU\\_MESR\\_MECCR\)](#) bits DEDD and CEDD, respectively. The disabled state turns ineffective any action due to error detection, including error correction. Note that error detection is always enabled out of reset.

#### 35.15.13.2.2 Memory error flags

SDM single (correctable) and non-correctable bit errors are flagged in [eTPU Memory Error Status Register \(eTPU\\_MESR\\_MESR\)](#) fields DCERR and DNCERR, respectively.

SCM correctable and non-correctable errors are flagged in the eTPU\_MESR\_MESR fields CCERR and CNCERR, respectively.

Double bit errors are guaranteed to be flagged, while three or more simultaneous bit errors in an SDM byte or SCM word are not. An error flagged as non-correctable can be in two or more bits.

Overflow flags DCOVR and CCOVR indicate, for SDM and SCM respectively, when a correctable error occurs when DCERR, CCERR are already asserted. Similarly, overflow flags DNCOVR/CNCOVR assert when a non-correctable error occurs when DNCERR/CNCERR are asserted already.

### 35.15.13.2.3 Memory error reports

The address, syndrome and data of a non-correctable SDM access can be sampled into [eTPU Data Error Report Address Register \(eTPU\\_MESR\\_DERAR\)](#), [eTPU Data Error Report Syndrome Register \(eTPU\\_MESR\\_DERSR\)](#), and [eTPU Data Error Report Data Register \(eTPU\\_MESR\\_DERDR\)](#), respectively. eTPU\_MESR\_DERAR also captures the SDM read access originator, whether the host, CDC, or specific eTPU channel service.

Similarly, non-correctable SCM accesses address, syndrome and data are stored in [eTPU Code Error Report Address Register \(eTPU\\_MESR\\_CERAR\)](#), [eTPU Code Error Report Syndrome Register \(eTPU\\_MESR\\_CERSR\)](#), and [eTPU Code Error Report Data Register \(eTPU\\_MESR\\_CERDR\)](#), respectively. eTPU\_MESR\_CERAR also captures the SCM read access originator, whether the host, MISC, or specific eTPU channel service. The occurrence of a non-correctable error overwrites the report register values from a correctable error, either on SDM (DCERR=1) or SCM (CCERR=1).

The last address, syndrome and data of a correctable SDM access can also be sampled into eTPU\_MESR\_DERAR, eTPU\_MESR\_DERSR, and eTPU\_MESR\_DERDR, but only if [DNCERR](#) is 0. That prevents access address, syndrome and data from a correctable error to overwrite the same information from a non-correctable access.

Similarly, the last address, syndrome and data from a correctable SCM access are stored in eTPU\_MESR\_CERAR, eTPU\_MESR\_CERSR, and eTPU\_MESR\_CERDR only if [CNCERR](#) is 0.

For either SDM or SCM, a new correctable error overwrites the report registers' values from an earlier correctable error, but report values from a non-correctable error are not overwritten, as long as bit DNCERR/CNCERR is asserted, neither by a new correctable error nor by a non-correctable one.

The sampling of SDM and SCM error report registers can be disabled through the eTPU\_MESR\_MECR fields DRDIS and CRDIS, respectively. When DRDIS (CRDIS) is asserted, the update of the SDM (SCM) error report registers is disabled, but not of

eTPU\_MESR\_MESR fields DCERR, DNCERR (CCERR, CNCERR). DRDIS, CRDIS must be set before reading the respective report registers to assure their coherent reading, preventing them to be updated between reads.

#### 35.15.13.2.4 Error detection and wait-states

Microengine SDM accesses do not take any wait-state due to ECC support. Host SDM read accesses take two wait-states minimum<sup>39</sup> in MCUs with SDM ECC support, regardless if an error was detected or not, if the SDM error detection is enabled ([eTPU Memory Error Control Register \(eTPU\\_MESR\\_MECR\)](#) field DEDD=0). Host SDM read accesses take a minimum of one wait-state if the SDM error detection is disabled (eTPU\_MESR\_MECR[DEDD]=1). Host SDM write accesses take one wait-state minimum, regardless of ECC support presence or enabling.

Host SCM read accesses have two wait-states<sup>39</sup>, regardless if an error was detected or not. Writes from the host to the SCM take one wait-state on 32-bit accesses, and one additional wait-state (two wait-states total<sup>39</sup> on 16-bit and 8-bit wide accesses. Those figures are not affected by the SCM error detection disabling (eTPU\_MESR\_MECR[CEDD]).

#### 35.15.13.2.5 Memory error interrupts

For **microengine accesses**, the Global Exception interrupt asserts whenever a non-correctable error is detected, either on SCM or SDM (see [Interrupts and data transfer requests](#)). These errors are flagged by [eTPU Module Configuration Register \(eTPU\\_SCR\\_MCR\)](#) fields SCMERR and SDMERR, respectively. Single bit errors are corrected and activate the Memory Error Correction Interrupt if it is enabled (see [Interrupt types and sources](#)), but do not generate Global Exceptions.

For **host accesses** a bus error is issued whenever a non-correctable error is detected, either on SCM or SDM (but not through the CDC) and it also generates a Global Exception. Single bit errors are corrected (if the error correction feature is available) and do not cause bus transfer errors, but assert the Memory Error Correction Interrupt if it is enabled (see [Interrupt types and sources](#)).

#### 35.15.13.2.6 Effect of errors on microcode execution

Correctable errors, either on SCM or SDM, do not affect microcode execution if detection/correction is enabled.

The executing thread is affected by a non-correctable SDM error as follows:

39. One wait-state more, compared with eTPU versions before eTPU2 and previous eTPU2 without ECC support



- If **DTEND** = 1: on SDM errors due to a read operation, if the instruction is not also an END, the following instruction automatically executes an END, and its operations may be executed, totally or partially. On SDM errors during the TST preload, the thread automatically ends on the first instruction, if the error occurs on a preload parameter selected by the entry point bit PP (see Entry Point Format section.) The first instruction may be executed totally or partially. Data read into P or DIOB may be corrupted in either case.
- If **DTEND** = 0: the TST or thread continues the execution flow normally. Data read into P or DIOB may be corrupted in either case.

The executing thread is affected by a non-correctable SCM error as follows:

- If **CTEND** = 1: on SCM errors, the corrupted instruction is not executed, instead replaced by a simple END. Due to instruction prefetching, an error may be detected and yet not affect the thread execution. However, an error detected in a prefetched instruction may cause error report and global or memory error interrupts, even if not executed. If a non-correctable error occurs on the entry table read during TST (see Time Slot Transition section), the first instruction in the thread is replaced with a simple END, a global exception is issued and the reports and flags are normally updated.
- If **CTEND** = 0: the execution flow continues normally, and the results depend on the particular instruction corruption.

**Table 35-418. Error support summary if CEDD = DEDD = 0**

| Action                            |            | SCM                                                           |                                                    | SDM              |                                              |
|-----------------------------------|------------|---------------------------------------------------------------|----------------------------------------------------|------------------|----------------------------------------------|
|                                   |            | single bit error                                              | double bit error                                   | single bit error | double bit error                             |
| Memory Error Correction Interrupt |            | Asserts                                                       | Not asserted                                       | Asserts          | Not asserted                                 |
| Global Exception                  |            | Not asserted                                                  | Asserts                                            | Not asserted     | Asserts                                      |
| Bus Error (host reads)            |            | Not issued                                                    | Issued                                             | Not issued       | Issued                                       |
| Thread                            | CTEND = 0  | Normal execution                                              | Potential erroneous execution <sup>1</sup>         | Normal execution | see DTEND                                    |
|                                   | CTEND = 1  |                                                               | Terminates before erroneous execution <sup>2</sup> |                  |                                              |
|                                   | DTEND = 0  |                                                               | see CTEND                                          |                  | Potential erroneous execution                |
|                                   | DTEND = 1  |                                                               |                                                    |                  | Terminates after erroneous read <sup>3</sup> |
| Code Error Report Registers       | CNCERR = 0 | Updates eTPU_MESR_CERAR, eTPU_MESR_CERSR, eTPU_MESR_CERDR     |                                                    | Not related      |                                              |
|                                   | CNCERR = 1 | eTPU_MESR_CERAR, eTPU_MESR_CERSR, eTPU_MESR_CERDR not updated |                                                    |                  |                                              |

Table continues on the next page...

**Table 35-418. Error support summary if CEDD = DEDD = 0 (continued)**

| Action                      |            | SCM              |                  | SDM                                                          |                  |
|-----------------------------|------------|------------------|------------------|--------------------------------------------------------------|------------------|
|                             |            | single bit error | double bit error | single bit error                                             | double bit error |
| Data Error Report Registers | DNCERR = 0 | Not related      |                  | Updates eTPU_MESR_DERAR, eTPU_MESR_DESR, eTPU_MESR_DERDR     |                  |
|                             | DNCERR = 1 |                  |                  | eTPU_MESR_DERAR, eTPU_MESR_DESR, eTPU_MESR_DERDR not updated |                  |

1. The thread executes correctly if the error is detected on an instruction fetched but not executed.
2. The erroneous instruction is not executed at all, and replaced by a simple END.
3. The instruction following the erroneous read may be executed totally or partially.

### 35.15.13.2.7 Error Correction Code (ECC) and syndrome definition

SDM ECC is composed of 5 Hsiao code<sup>40</sup> bits for each data byte. SCM ECC is composed of 7 Hsiao code bits for each 32-bit word.

Table 35-419 shows the encoding of the SDM syndrome and global parity fields as they read on eTPU Data Error Report Syndrome Register (eTPU\_MESR\_DESR). There is one syndrome and one global parity for each byte in an SDM word, numbered SYND0 to SYND3 from the least to the most significant.

**Table 35-419. SDM syndrome definition**

| SYNDn<br>(hex) | Result                 | SYNDn<br>(hex) | Result                        |
|----------------|------------------------|----------------|-------------------------------|
| 0E             | error on data bit 8n+7 | 10             | error on <b>code</b> bit 8n+4 |
| 15             | error on data bit 8n+6 | 08             | error on <b>code</b> bit 8n+3 |
| 07             | error on data bit 8n+5 | 04             | error on <b>code</b> bit 8n+2 |
| 1A             | error on data bit 8n+4 | 02             | error on <b>code</b> bit 8n+1 |
| 19             | error on data bit 8n+3 | 01             | error on <b>code</b> bit 8n   |
| 13             | error on data bit 8n+2 |                |                               |
| 16             | error on data bit 8n+1 | 00             | no error                      |
| 1C             | error on data bit 8n   | any other      | non-correctable error         |

The following table shows the definition of the SCM syndrome field as it reads on eTPU Code Error Report Syndrome Register (eTPU\_MESR\_CERSR).

40. Hsiao, M.Y.; A Class of Optimal Minimum Odd-weight-column SEC-DED Codes; IBM Journal of Research and Development [http://ieeexplore.ieee.org/xpl/freeabs\\_all.jsp?arnumber=5391627](http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5391627)



Table 35-420. SCM syndrome definition

| SYND<br>(hex) | Result                     | SYND<br>(hex) | Result               | SYND<br>(hex) | Result                |
|---------------|----------------------------|---------------|----------------------|---------------|-----------------------|
| 40            | error on <b>code</b> bit 6 | 58            | error on data bit 25 | 46            | error on data bit 11  |
| 20            | error on <b>code</b> bit 5 | 0B            | error on data bit 24 | 4A            | error on data bit 10  |
| 10            | error on <b>code</b> bit 4 | 54            | error on data bit 23 | 52            | error on data bit 9   |
| 08            | error on <b>code</b> bit 3 | 15            | error on data bit 22 | 62            | error on data bit 8   |
| 04            | error on <b>code</b> bit 2 | 16            | error on data bit 21 | 13            | error on data bit 7   |
| 02            | error on <b>code</b> bit 1 | 34            | error on data bit 20 | 29            | error on data bit 6   |
| 01            | error on <b>code</b> bit 0 | 25            | error on data bit 19 | 31            | error on data bit 5   |
|               |                            | 26            | error on data bit 18 | 43            | error on data bit 4   |
| 49            | error on data bit 31       | 64            | error on data bit 17 | 45            | error on data bit 3   |
| 0D            | error on data bit 30       | 2C            | error on data bit 16 | 19            | error on data bit 2   |
| 0E            | error on data bit 29       | 1A            | error on data bit 15 | 51            | error on data bit 1   |
| 38            | error on data bit 28       | 23            | error on data bit 14 | 61            | error on data bit 0   |
| 4C            | error on data bit 27       | 2A            | error on data bit 13 | 00            | no error              |
| 1C            | error on data bit 26       | 32            | error on data bit 12 | any other     | non-correctable error |

### 35.15.13.2.8 Error injection

Errors can be injected into the SDM read data path through [eTPU Data Error Injection Address Register \(eTPU\\_MESR\\_DEIAR\)](#), [eTPU Data Error Injection Data Pattern Register \(eTPU\\_MESR\\_DEIDPR\)](#) and [eTPU Data Error Injection Parity Pattern Register \(eTPU\\_MESR\\_DEIPPR\)](#). They define bit flips on accesses at the specific address specified in eTPU\_MESR\_DEIAR.

SDM error injection is enabled independently for host and microengine accesses by the [eTPU Module Configuration Register \(eTPU\\_SCR\\_MCR\)](#) fields HDEIE and MDEIE, respectively. The user must program eTPU\_MESR\_DEIAR, eTPU\_MESR\_DEIDPR, and eTPU\_MESR\_DEIPPR with HDEIE=MDEIE=0, and only then enable error injection by setting HDEIE=1 and/or MDEIE=1 to avoid races (see [eTPU Memory Error Status Register \(eTPU\\_MESR\\_MESR\)](#)).

Similarly, SCM errors can be injected into the SCM read data path through [eTPU Code Error Injection Address Register \(eTPU\\_MESR\\_CEIAR\)](#), [eTPU Code Error Injection Data Pattern Register \(eTPU\\_MESR\\_CEIDPR\)](#) and [eTPU Code Error Injection Parity Pattern Register \(eTPU\\_MESR\\_CEIPPR\)](#), enabled by the [eTPU Module Configuration Register \(eTPU\\_SCR\\_MCR\)](#) field CEIE. Errors are injected on host accesses to the address specified in eTPU\_MESR\_CEIAR when **VIS** =1, or on microengine and MISC accesses when **VIS**=0 .

### 35.15.13.2.9 Error fixing

Single bit **soft errors** can be fixed by rewriting the right value, as corrected by the ECC logic, into the same address of the erroneous data. eTPU provides means to perform such fixes safely, coherently and concurrently with the normal operation.

**SDM fixes** can be done using the CDC (see [Coherent Dual-parameter Controller \(CDC\)](#)), limited to the lower 8 Kbytes of SDM. The Host application simply has to program a transfer with both origin and destination positions pointing to the address of the data to be fixed. Because the PBBASE field has an address granularity of 8 bytes, a word adjacent to the one to be fixed must also be rewritten. The CDC transfer is atomic, therefore the fix is done coherently and does not interfere with the application code or data flows.

**SCM fixes** can be done by programming [eTPU Code Error Fix Register \(eTPU\\_MESR\\_CEFR\)](#) (see [eTPU Code Error Report Syndrome Register \(eTPU\\_MESR\\_CERSR\)](#)). The fixes are performed by the MISC logic, which must be enabled (see [SCM Test - Multiple Input Signature Calculator](#)). The modes available, selected by [CFIXM](#), are:

- **Automatic Fix Mode (CFIXM=11)**: the MISC logic automatically rewrites the contents of an address whenever it finds a correctable error; the CFIXM field keeps its value if a fix is made.

## 35.15.14 Performance monitoring features

### 35.15.14.1 Idle Counter

The Idle Counters [eTPU Idle Register for Engine 1 \(eTPU\\_ER\\_IDLER\\_ENG1\)](#) and [eTPU Idle Register for Engine 2 \(eTPU\\_ER\\_IDLER\\_ENG2\)](#) continuously count microcycles in which the microengine is not busy with channel service. They can be used to measure the microengine utilization by rating the count measured during a period of time to the number of microcycles contained in the period. The Idle Counters do not count microcycles when the engine is stopped, or is in TST or halt states.

### 35.15.14.2 Latency Monitor

eTPU implements Latency Monitors to measure the time taken to service a request. Each Latency Monitor provides:

- The measure of the greatest latency found during actual thread executions, accessed from [eTPU Latency Monitor Measure Register for Engine 1](#) (`eTPU_LMR_LMnMR_ENG1`) and [eTPU Latency Monitor Measure Register for Engine 2](#) (`eTPU_LMR_LMnMR_ENG2`).
- A comparison between the measured latency and a threshold, determined through [eTPU Latency Monitor Measure Register for Engine 2](#) (`eTPU_LMR_LMnMR_ENG2`) and [eTPU Latency Monitor Configuration Register for Engine 2](#) (`eTPU_LMR_LMnCR_ENG2`).

Each Latency Monitor works independently in one of the following modes, selected with the LMM field in [eTPU Latency Monitor Configuration Register for Engine 1](#) (`eTPU_LMR_LMnCR_ENG1`) and [eTPU Latency Monitor Configuration Register for Engine 2](#) (`eTPU_LMR_LMnCR_ENG2`) :

- **Disabled mode:** no latency measures or actions are taken.
- **Channel mode:** monitors the latency of an individual channel.
- **Priority mode:** monitors the combined latency of all channels in a given priority level - High, Medium or Low (see [Scheduler](#)).

Channel and Priority modes are detailed in the subsections below. The following applies to both modes:

- After each latency measurement, the value measured is compared with the LATENCY value in [eTPU Latency Monitor Measure Register for Engine 1](#) (`eTPU_LMR_LMnMR_ENG1`) and [eTPU Latency Monitor Measure Register for Engine 2](#) (`eTPU_LMR_LMnMR_ENG2`). If the new measured value is greater, it replaces the old value in [eTPU Latency Monitor Measure Register for Engine 1](#) (`eTPU_LMR_LMnMR_ENG1`) and [eTPU Latency Monitor Measure Register for Engine 2](#) (`eTPU_LMR_LMnMR_ENG2`).
- After each latency measurement, the value measured is compared with the MAXLAT value in [eTPU Latency Monitor Configuration Register for Engine 1](#) (`eTPU_LMR_LMnCR_ENG1`) and [eTPU Latency Monitor Configuration Register for Engine 2](#) (`eTPU_LMR_LMnCR_ENG2`), and if the new measured value is greater:

- A Global Exception is issued.
- The [eTPU Module Configuration Register \(eTPU\\_SCR\\_MCR\)](#) flag LTO1 (engine 1) or LTO2 (engine 2) is set, independently of the LATENCY value in [eTPU Latency Monitor Measure Register for Engine 1 \(eTPU\\_LMR\\_LMnMR\\_ENG1\)](#) and [eTPU Latency Monitor Measure Register for Engine 2 \(eTPU\\_LMR\\_LMnMR\\_ENG2\)](#).

**NOTE**

The LATENCY value is updated even if MAXLAT=0.

**NOTE**

MAXLAT=0 just inhibits the assertion of the LTO1/2 flag and Global Exception due to the respective Latency Monitor. Other Latency Monitor instances with non-zero MAXLAT value may assert LTO1/2 and Global Exception.

**35.15.14.2.1 Channel mode**

When the Latency Monitor operates in Channel mode:

- Only the channel selected by the MONCHAN field in [eTPU Latency Monitor Configuration Register for Engine 1 \(eTPU\\_LMR\\_LMnCR\\_ENG1\)](#) and [eTPU Latency Monitor Configuration Register for Engine 2 \(eTPU\\_LMR\\_LMnCR\\_ENG2\)](#) is monitored.
- Any combination of Host Service Requests (HSR), Channel Service Requests and Link Service Requests can be selected for monitoring, based on the SRSEL field in [eTPU Latency Monitor Configuration Register for Engine 1 \(eTPU\\_LMR\\_LMnCR\\_ENG1\)](#) and [eTPU Latency Monitor Configuration Register for Engine 2 \(eTPU\\_LMR\\_LMnCR\\_ENG2\)](#).
- The latency is counted in microcycles, from the moment the logic OR of all the service requests selected by SRSEL asserts to the moment this logic OR negates.
- The latency count is interrupted when the channel is disabled, either by programming the CPR field in [eTPU Channel n Configuration Register for Engine 1 \(eTPU\\_CCCR\\_CnCR\\_ENG1\)](#) and [eTPU Channel n Configuration Register for Engine 2 \(eTPU\\_CCCR\\_CnCR\\_ENG2\)](#) to 0, or by a watchdog timeout. Latency is counted until the channel is disabled and is not discarded; a new count can start from zero when the channel is re-enabled.

**NOTE**

If a selected service request type or a combination of them continues asserted during and after its service thread execution,

its latency counter will keep on running; this can happen, for instance, if either matches can service requests (e.g. channel modes `em_st/em_dt`), one match happens before the other is serviced, and only one of them is cleared. The Latency Monitor can, in cases like this, yield pessimistic measurements.

### NOTE

The Latency Monitor in channel mode must be disabled before the channel it monitors is disabled.

#### 35.15.14.2.2 Priority mode

When the Latency Monitor operates in Priority mode:

- Only the combination of channels assigned with the priority selected by the `SRSEL` field in [eTPU Latency Monitor Configuration Register for Engine 1](#) (`eTPU_LMR_LMnCR_ENG1`) and [eTPU Latency Monitor Configuration Register for Engine 2](#) (`eTPU_LMR_LMnCR_ENG2`) is monitored.
- The latency is measured in microcycles, starting from either of these two events, whichever happens first:
  - The logic OR of service requests of the selected priority transitions from 0 to 1 at a moment when all their grants are at state negated.
  - The grants of the selected priority are cleared by the scheduler at the end of the thread, and any of the service requests at the same priority is still active soon after the thread ends.
- The latency is measured until the grants of the selected priority are cleared by the scheduler, after the measurement started.

### Note

A Latency Monitor in priority mode must be disabled before any of the channels in the selected priority is disabled, or any disabled channel is assigned with the selected priority.

### 35.15.14.3 Watchdog Monitor Operation

The watchdog mechanism can also be used to monitor, without interfering with the application execution, whether the execution time of any individual thread or a sequence of threads reaches a given limit. This can be achieved by asserting the WMO field in [eTPU Watchdog Timer Register for Engine 1 \(eTPU\\_ER\\_WDTR\\_ENG1\)](#) and [eTPU Watchdog Timer Register for Engine 2 \(eTPU\\_ER\\_WDTR\\_ENG2\)](#).

## 35.16 Initialization/application information

### 35.16.1 Configuration sequence

After initial power-on reset the eTPU remains in an idle state<sup>41</sup>, requiring initialization of several registers before any function can begin execution. Also, if the SCM is implemented in RAM, it should be initialized with the eTPU application code prior to configuring the eTPU. Configuration procedures are summarized as follows:

- If SCM is implemented as RAM, load the eTPU application code (see [SCM access](#)).
- Initialize SDM (optional if no ECC support available)
- Initialize the SCM MISC logic (optional, see [SCM Test - Multiple Input Signature Calculator](#)).
- Initialize Watchdog and Latency Monitors (optional).
- Initialize [eTPU Engine 1 Time Base Configuration Register \(eTPU\\_TBR\\_TBCR\\_ENG1\)](#) and [eTPU Engine 2 Time Base Configuration Register \(eTPU\\_TBR\\_TBCR\\_ENG2\)](#) to set up:
  - TCR1 and TCR2 prescalers and clock sources.
  - Select digital filtering mode.
  - TCRCLK signal filter control.
  - Angle mode operation (if necessary).
- Initialize [eTPU Engine 1 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG1\)](#) and [eTPU Engine 2 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG2\)](#) to set up:

---

41. Except when device debug request is asserted on power-on reset: in this case, the microengines wake-up in halt state.

- Entry table base.
- Filter prescaler clock control.
- Initialize eTPU Engine 1 STAC Configuration Register (eTPU\_TBR\_STAC\_ENG1) and eTPU Engine 2 STAC Configuration Register (eTPU\_TBR\_STAC\_ENG2) if you need to set up TCR1/2 resource Client/Server operation.
- Write to eTPU Channel n Configuration Register for Engine 1 (eTPU\_CCCR\_CnCR\_ENG1) and eTPU Channel n Configuration Register for Engine 2 (eTPU\_CCCR\_CnCR\_ENG2) to choose the Function to be performed by each channel, and its parameter base address.
- Write to eTPU Channel n Status Control Register for Engine 1 (eTPU\_CCCR\_CnSCR\_ENG1) and eTPU Channel n Status Control Register for Engine 2 (eTPU\_CCCR\_CnSCR\_ENG2) to choose among the possible variations within the function flow (FM bits).
- Write to SDM for parameter initialization of each configured channel.
- Write to eTPU Watchdog Timer Register for Engine 1 (eTPU\_ER\_WDTR\_ENG1) and eTPU Watchdog Timer Register for Engine 2 (eTPU\_ER\_WDTR\_ENG2) if you need to enable and set up the Watchdog(s) mode and timeout.
- Write to eTPU Channel n Host Service Request Register for Engine 1 (eTPU\_CCCR\_CnHSRR\_ENG1) and eTPU Channel n Host Service Request Register for Engine 2 (eTPU\_CCCR\_CnHSRR\_ENG2) to initialize the active channels.
- Write to eTPU Channel Interrupt Enable Register for Engine 1 (eTPU\_GCR\_CIER\_ENG1) and eTPU Channel Interrupt Enable Register for Engine 2 (eTPU\_GCR\_CIER\_ENG2) if interrupts are to be enabled from the appropriate channels. Likewise for Data Transfer Requests (eTPU Channel Data Transfer Request Enable Register for Engine 1 (eTPU\_GCR\_CDTRER\_ENG1) and eTPU Channel Data Transfer Request Enable Register for Engine 2 (eTPU\_GCR\_CDTRER\_ENG2)). This can also be done through eTPU Channel n Configuration Register for Engine 1 (eTPU\_CCCR\_CnCR\_ENG1) and eTPU Channel n Configuration Register for Engine 2 (eTPU\_CCCR\_CnCR\_ENG2).
- Write to eTPU Channel n Configuration Register for Engine 1 (eTPU\_CCCR\_CnCR\_ENG1) and eTPU Channel n Configuration Register for Engine 2 (eTPU\_CCCR\_CnCR\_ENG2) to enable each channel by assigning it a high, middle, or low priority (CPR field).<sup>41</sup>



- Monitor [eTPU Channel n Host Service Request Register for Engine 1 \(eTPU\\_CCCR\\_CnHSRR\\_ENG1\)](#) and [eTPU Channel n Host Service Request Register for Engine 2 \(eTPU\\_CCCR\\_CnHSRR\\_ENG2\)](#) for completion of initialization.
- Write [GTBE](#) =1 to start TCR1/TCR2 time base counting at same time in both Engines (may be done before or never, depending on the particular application and use of Red Line bus).

See [Initialization Code Example](#).

## 35.16.2 Reset options

### 35.16.2.1 Hardware reset

Hardware reset is achieved by assertion of device synchronous reset. Both Engines and common logic is reset, and even the System Configuration and Global Channel registers assume their reset values.

#### Note

All eTPU input clocks must pulse during reset so that both engines are reset, even if they are in Module Disable or Stop mode.

### 35.16.2.2 Software reset

eTPU has no software reset. To abort infinite microcode loops, the Force END mechanism must be used (see field FEND in [eTPU Engine 1 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG1\)](#) and [eTPU Engine 2 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG2\)](#)).

## 35.16.3 Multiple parameter coherency methods

Follows a description of two methods for coherent transfer of multiple parameters between Host and eTPU. Both methods involve the use of two parameter areas: the Transfer Parameter Area (hereafter called TPA), which is the SDM area directly accessed by the Host for reads and writes, and the Permanent Parameter Area (hereafter called



PPA), which are the SDM positions where channel parameters are normally accessed by the Function microcode. Note that parameters in either TPA or PPA do not have to be in sequential addresses. TPAs and PPAs allocation are completely defined by the application, and there may be any number of them, independently of the channels.

The methods described here are not the only solutions for the coherent transfer problem, and both can coexist in eTPU and even used in combination. Also note that for transfers of a pair of parameters, the Coherent Dual-parameter Controller is faster and have less impact on both eTPU and Host performance. That said, the methods are:

- **Transfer Service:** a microengine thread transfers, upon Host Service Request, data from/to a TPA to/from a PPA. Coherency is guaranteed by the fact that a thread is atomic with respect to other threads in the same Engine, and so are its transfers. If parameters in PPA are shared by both Engines, hardware semaphores have to be used to access them.
- **Mailbox:** for Host to eTPU transfers, the microcode checks a flag, set by the host, indicating the existence of new parameter data in the TPA. It can, then, either access TPA data directly or copy it to the PPA. For eTPU to Host transfers, when microcode changes PPA, it copies them to the TPA and flags updated TPA data to Host, possibly using an Interrupt or a Data Transfer Request. The Mailbox flag is reset when data is copied: by the eTPU microcode, when it transfers TPA to PPA (possibly followed by an Interrupt); by the Host, when it reads data from the TPA. This indicates that TPA is free for another transfer.

Transfer Service has the advantage of separating the task of data transfer from the functional service thread that accesses the parameters, with less impact to the latter. Compared to the Mailbox method, however, it has bigger average latency, because the Transfer Service thread has to contend for a time slot to execute. This latency can be minimized if Transfer Service thread is assigned to a separate channel with higher priority, but even so it does not guarantee that PPA is updated before the next execution of the functional thread that uses it.

Mailbox method, on the other hand, makes the functional thread check for the existence of new data (Host to eTPU). It does not have to be responsible for the transfer, though: it may access the TPA directly, and a Transfer Service can then be used to copy data from TPA to PPA.

## 35.16.4 Programming Hints and Caveats

### 35.16.4.1 Atomic dual access after a call, return

A dual, back-to-back parameter access is not atomic after a call, a jump, or a return if they occurred in parallel with an odd SDM access. It is safer to make a pair of parameter accesses that must be coherent begin at the second instruction after a call/jump/return.

### 35.16.4.2 Resource polling

The use of polling while waiting for a condition or a resource (except semaphore lock) should be avoided in order not to hang the microengine in long loops. This general programming guideline is greatly enforced in eTPU, as a thread cannot be preempted for any reason. Safer polling, albeit with long and indeterministic latency, can be obtained if a thread issues a channel link to the serviced channel and terminates. The microengine is then free to other tasks, and the next poll happens at the next time the channel is serviced. This mechanism can be combined with finite (timed out) loops for better latency.

### 35.16.4.3 Changing channel function, parameter base, or entry table scheme

Channel function, parameter base address and entry table scheme are determined by the CFS, CPBA, and ETCS fields in [eTPU Channel n Configuration Register for Engine 1 \(eTPU\\_CCCR\\_CnCR\\_ENG1\)](#) and [eTPU Channel n Configuration Register for Engine 2 \(eTPU\\_CCCR\\_CnCR\\_ENG2\)](#). They cannot be changed when the channel is enabled. If the channel is disabled first, one may still have service requests from the previous function, so before the channel is enabled again one must be sure that:

- the first thread executed in the new function is the initialization one.
- the initialization thread of the new function clears any previously pending service request.

Here is a safe procedure for function changing:

1. Disable the channel (in [eTPU Channel n Configuration Register for Engine 1 \(eTPU\\_CCCR\\_CnCR\\_ENG1\)](#) and [eTPU Channel n Configuration Register for Engine 2 \(eTPU\\_CCCR\\_CnCR\\_ENG2\)](#), write CPR=00).
2. Change the function configuration (fields CFS and/or CPBA and/or ETCS in [eTPU Channel n Configuration Register for Engine 1 \(eTPU\\_CCCR\\_CnCR\\_ENG1\)](#) and [eTPU Channel n Configuration Register for Engine 2 \(eTPU\\_CCCR\\_CnCR\\_ENG2\)](#)).

3. Request the initialization thread, writing [eTPU Channel n Host Service Request Register for Engine 1 \(eTPU\\_CCCR\\_CnHSRR\\_ENG1\)](#) and [eTPU Channel n Host Service Request Register for Engine 2 \(eTPU\\_CCCR\\_CnHSRR\\_ENG2\)](#) with the initialization HSR (channel still disabled).
4. enable the channel (in [eTPU Channel n Configuration Register for Engine 1 \(eTPU\\_CCCR\\_CnCR\\_ENG1\)](#) and [eTPU Channel n Configuration Register for Engine 2 \(eTPU\\_CCCR\\_CnCR\\_ENG2\)](#) write  $CPR > 0$ ); the initialization HSR is serviced before any other formerly pending service requests, clearing them.

#### 35.16.4.4 Checking and Clearing Interrupts of a Stopped Engine

An Engine may be stopped with interrupts (or DMA requests) pending. This includes the case when the Engine's MDIS bit is set and a thread is still running: the thread will complete execution, possibly issuing an interrupt or DMA request before the engine stops, setting the STF bit.

As soon as the engine stops the channel registers become inaccessible, issuing bus errors when accessed. Interrupts and DMA requests can still be checked and cleared through the Global Channel Registers, though. DMA requests can also be cleared by the hardware handshaking with the DMA controller when the engine is stopped.

#### 35.16.5 Estimating Worst Case Latency

Reliable systems are designed to work under worst-case conditions. This section explains how to estimate worst-case latency (WCL) for any eTPU function in any system. The appendix covers the following topics:

- Introduction to Worst-Case Latency
- Using Worst-Case Latency Estimates to Evaluate Performance
- Priority Scheme Details used in WCL Analyses
- First-Pass WCL Analysis
- Second-Pass WCL Analysis

The first-pass WCL analysis is based on a deterministic, generalized formula that is easy to apply. Because of the generalizations in the formula, the first analysis result is almost always much worse than the real worst case. If the desired system performance is within the limits of this first analysis, then no further analysis is required; the system is well

within the performance limits of the eTPU. If the desired system performance exceeds that indicated by the first analysis, the second-pass WCL analysis should be applied. The second-pass analysis is not a generalized formula, but rather uses specific system details for a realistic worst-case estimation.

### 35.16.5.1 Introduction to Worst-Case Latency

#### Note

The latency calculation and examples refer to old TPU functions such as PWM, DIO etc. These functions use single action channels which have single transition and single match functionality. They are not optimized for the eTPU hardware enhancement which support various double action modes. These examples are for reference only. New eTPU functions which are optimized for the new hardware will impose different latency calculations.

Worst-case latency for a channel is the longest amount of time that can elapse between the execution of any two function threads on that channel. For example, if in a particular system, channel 5 is running PWM, the worst-case latency for channel 5 is the longest possible time between the execution of two PWM threads. The worst case time includes the time the execution unit takes to execute threads for other active channels, and other delays described later in this section. Refer to the following figure.

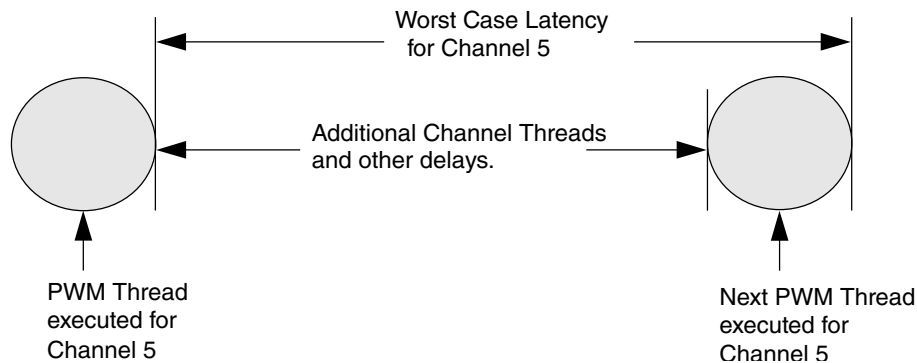


**Figure 35-335. Worst-Case Latency for PWM**

Worst-case latency for a channel depends both on the function running on that channel and on the activity on other channels. Since the 32 eTPU channels must all share the same execution unit, execution speed of a particular function varies with each system. The PWM thread response is faster if there are no other active channels than if other

channels are also active. In addition, changing the priority scheme and channel number assignments can change performance for a function even if the same set of functions are still active.

Each function is divided into threads, as shown in the figure below (see also [Functions and threads](#)). The eTPU Microengine executes one thread of a function at a time. For example, the Microengine might execute thread 1 of PWM, then thread 3 of DIO, then thread 2 of PWM, then thread 2 of SM, and so on. The amount of time the eTPU Microengine grants a function to execute a thread varies with the number of microcode instructions in the thread.

Since there is only one eTPU Microengine (in each eTPU Engine), the eTPU cannot actually execute the software for multiple functions simultaneously. However, the hardware for each of the channels is independent. This means that, for example, all 32 channel signals can change thread at the same moment, provided that the function software sets up the channel hardware to do so beforehand.

With Host CPU code, the system designer assigns functions to channels and initializes the functions. After initialization, functions typically run without Host intervention, except for eTPU channel interrupts to the Host to give or receive information. Most functions can run continuously with periodic servicing from the eTPU Microengine. As required, the channels request service from the eTPU Microengine, and the eTPU Scheduler determines the order in which the channels are serviced. Worst-case latency for a channel can be derived from the details of the priority scheme that the scheduler uses (see [Scheduler](#)).

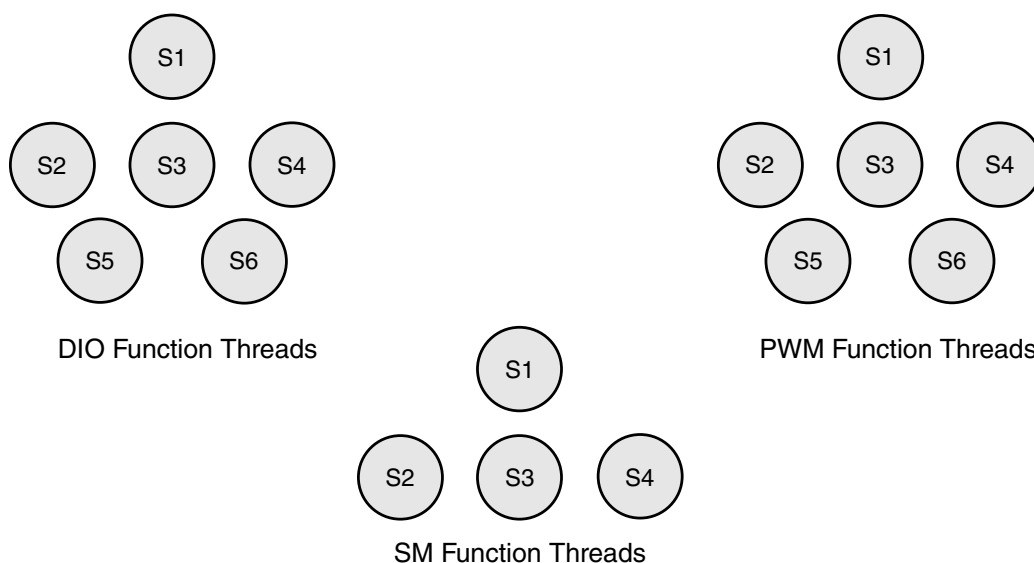


**Figure 35-336. Function Threads**

### 35.16.5.2 Using Worst-Case Latency estimates to evaluate performance

Once WCL is found for a channel, the user must determine how to use this number to analyze performance. To analyze the performance of a channel running the PWM function, for example, some information about what happens in each thread is necessary.

The following example refers to old TPU PWM function, which is not optimized to the eTPU enhanced hardware. For PWM, thread 1 is the initialization thread, and threads 2 and 3 are used during normal function execution. (PWM threads 4, 5, and 6 are for special modes and will be assumed to be unused on channel 5). Thread 2 writes a time into the channel 5 match register and performs other operations that will cause the channel 5 signal to go from low to high at the time indicated in the match register (match time). At match time, the signal goes high and channel 5 requests service from the eTPU Microengine to execute thread 3. Thread 3 writes a time into the channel 5 match register and performs other operations that will cause the channel 5 signal to go from high to low at match time. At match time, the signal goes low and channel 5 requests service from the eTPU Microengine to execute thread 2. A PWM wave is kept running on the system by the eTPU executing thread 2, then thread 3, then thread 2, then thread 3, and so on.

Since the definition of worse-case latency assumes a fully loaded running system, initialization threads are not part of worst-case calculations. For the channel 5 example, the two PWM threads in [Figure 35-335](#) are thus the two normal running threads, threads 2 and 3.

[Figure 35-335](#) does not define which thread is thread 2 and which is thread 3. Since the worst-case latency derived from the first-pass analysis is the worst case between any 2 threads (not counting initialization threads), it is safe to say that the worst-case latency shown in [Figure 35-336](#) represents both the worst-case high time and the worst-case low time.

Notice in [Figure 35-335](#) that worst-case latency is drawn from the end of the execution of the first PWM thread to the end of the execution of the next PWM thread. It is drawn from end to end because the microcode instructions that make up the threads control the channel hardware. To make sure that all the microcode instructions needed to change the pin thread have been executed, it is necessary to include the execution time of the second thread.

Thread information for each function is found in the programming notes for individual TPU functions. See NXP Programming Note TPUPN00/D, Using the TPU Function Library and TPU Emulation Mode, for a list of available programming notes. Similar documentation will be provided for the eTPU new functions.

### 35.16.5.3 Priority Scheme Details Used in WCL Analysis

The user assigns functions to channel numbers and gives each active channel a priority level of high, middle, or low. The Scheduler uses the channel number and channel priority level to determine the order in which to grant service.

The scheduler allocates time slots to specific priority levels of high, middle, or low. One function thread is executed in each time slot. The length of a time slot varies according to the length of the executing thread. When fully loaded, the scheduler always assigns time slots in a seven-slot sequence (see the following table). After a seven-slot sequence is completed, another seven-slot sequence begins (see [Figure 35-338](#)). Note that in eTPU, when no service request exists, the scheduler goes to thread 1, but WCL calculation considers full load.

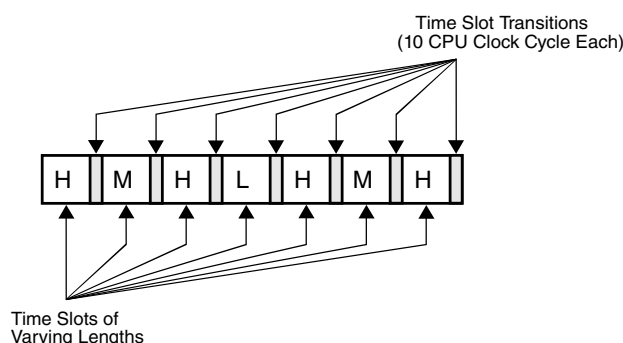


**Figure 35-337. Time-Slot Sequence**

This sequence scheme gives higher-priority channels more service time than lower-priority channels. High-priority channels are allocated four of seven time slots, middle-priority channels are allocated two of seven time slots, and low-priority channels are allocated one of seven time slots.

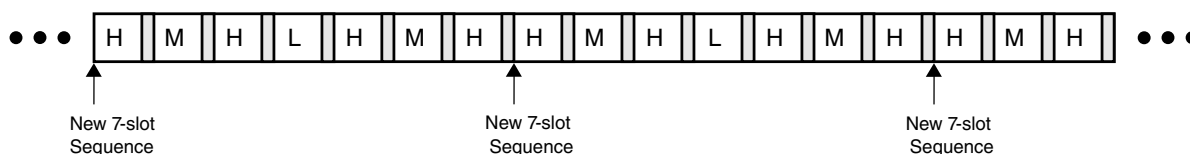


**Figure 35-338. Multiple Time-Slot Sequences**

#### 35.16.5.3.1 Priority Passing

If no channel of the priority level assigned to the time slot is requesting service, the eTPU scheduler can pass priority to other levels. If no high-level channel is requesting service during a high level time slot, a middle-level channel is granted service; or, if no middle level-channel is requesting service, a low-level channel is granted service. If no middle-level channel is requesting service during a middle-level time slot, a high-level channel is granted service; or, if no high-level channel is requesting service, a low-level channel is

granted service. If no low-level channel is requesting service during a low-level time slot, a high-level channel is granted service; or, if no high-level channel is requesting service, a middle-level channel is granted service. If no channel is requesting service, the time slot sequence is reset to state 1 and the scheduler idles until a request is received.

Priority passing is implemented in hardware and does not contribute to worst-case latency.

### 35.16.5.3.2 Time-slot transition

After each time slot, the eTPU must prepare for the next time slot. This preparation time between each time slot is called a time-slot transition. See [Time Slot Transition](#). Time-slot transitions can take from six up to ten eTPU clocks.

### 35.16.5.3.3 Channel Number Priority

If more than one channel of a priority level is requesting service, the lowest numbered channel is granted service first. For example, if channels 0, 5, and 9 are all high-level channels requesting service during a high time slot, channel 0 is granted service first. Continuing this example, if channel 0 requests service again immediately after being serviced, it is not serviced again until channels 5 and 9 are serviced. This scheme is implemented so that continuously-requesting low numbered channels do not take all the time on the eTPU execution unit and leave no time for other channels.

The scheduler uses registers to keep track of which channels have been serviced and which require servicing. Each channel has two register bit: a service request register (SRR) and a service grant register (SGR). The SRR is set when a channel requests service. After the channel has been granted service, the SGR is set and the SRR is cleared.

SGRs are not cleared individually by channel, but rather as priority level groups. The clearing of a group of SGRs begins a new cycle for that priority level. An SGR group is cleared on the condition that a channel of that priority level has just been serviced, and no other channel of that priority level is requesting service (has a set SRR) and has not been granted service (has a clear SGR).

For example, if a middle-priority channel has just been serviced (either in a middle-priority time slot or a high or low-priority time slot gained by priority passing), the SRRs and SGRs of all middle-priority channels are compared. If there is no middle-priority channel with its SRR set and SGR cleared, the scheduler clears all middle-level SGRs. If there is a middle-level channel with its SRR set and SGR cleared, the scheduler does not clear the SGR group, and the requesting middle-level channel is serviced on the next middle-level time slot (or possibly sooner by priority passing).



### 35.16.5.3.4 SDM Collision Rate

Most function threads read or write to the eTPU SDM at least once. Because both the eTPU Microengine and Host can access the SDM but not at the same time, the Microengine may suspend execution during the SDM access while waiting for the Host to finish accessing the SDM. At other times the Host may wait for the Microengine. Wait states can take up to two eTPU clocks, when the Host accesses the SDM directly, without using CDC. Microengine(s) wait-states must be added into the worst-case latency calculation. The system designer should estimate the percentage of SDM accesses in the system that will result in Microengine wait-states. This percentage is called the RAM collision rate (RCR). In each collision with direct Host accesses to the SDM the Microengine(s) wait for two eTPU clocks.

In eTPU the Coherent Dual-parameter Controller (CDC) may also access the SDM for atomic transfers of two parameters. eTPU Microengine may wait on this operation (if it is in service time) until the transfer is complete. CDC always transfers two parameters, making four consecutive accesses (read, write, read, write) of one eTPU clock each. The system designer should estimate the percentage of SDM accesses in the system that will result in a Microengine wait due to coherent transfer, and multiply it with the average number of eTPU clocks the Microengine waits for each transfer. This percentage is called Coherent Parameter Collision Rate (CPCR).

In addition, Microengine to Microengine multiple parameter coherent communication, using the hardware semaphores, may hold one Microengine which waits to lock the semaphore while the other Microengine is holding it. This waiting is due to a software loop, not hardware wait-states. Note that single parameter access of one Microengine does not affect the timing of the other Microengine due to SDM time interlace. This implies that single parameter Microengine to Microengine communication does not affect the performance. The Microengine which waits for the semaphore will loop until it is freed by the other Microengine. This time depends on the eTPU application. The system designer should estimate the percentage of Microengine to Microengine coherent parameter communication that will result in eTPU semaphore loops, and multiply it with the average number of eTPU clocks the Microengine loops for each such transfer. This percentage is called CCR (Communication Collision Rate).

A 100% collision rate for a system is the theoretical worst case. In many systems, however, the RCR, CPCR and CCR would be very low, sometimes even near 0%. This is because the eTPU is an independent processor capable of servicing most function needs, so that the Host rarely needs to access the eTPU SDM. Also coherent Microengine to Microengine communication of more than one parameter may be rare. To find a realistic

RCR, CPCR the system designer should evaluate the Host code and find the percentage of time it accesses the eTPU SDM with or without using the CDC. This percentage gives a good RCR and CPCR. The eTPU application provides a good estimation of CCR.

### Note

The programming practice of polling a flag in the eTPU SDM causes a very high RCR and should be avoided in high-performance systems.

After the collision rate for a system is found, it can be applied to the WCL calculations for each channel. The system designer can use the collision percentage and the number of SDM accesses (with and without semaphores) to estimate the eTPU loop time for a function. Note that in old TPU functions CPCR and CCR are both zero.

The estimation of eTPU wait time is as follows:

#### Variables:

$N1$  = Number of simple RAM accesses in the longest thread

$RCRWait$  = Maximal eTPU clocks wait time for simple RAM collision = 2

$CPCRWait$  = Average eTPU clocks for Coherent Parameter Transfer (using CDC).

$N2$  = Number of eTPU-eTPU semaphore RAM accesses in the longest thread

$CCRWait$  = Average eTPU clocks for Microengine-Microengine communication transfer.

#### Estimated Wait Time:

Function eTPU maximal wait time =

$$N1 * (RCR * RCRWait + CPCR * CPCRWait) + N2 * CCR * CCRWait$$

### 35.16.5.4 First-Pass Worst-Case Latency Analysis

Following is the first-pass calculation of worst-case latency for a channel. Remember that this analysis uses generalizations that usually produce a result much worse than the real worst case. If the worst-case result from the first analysis is too long for the desired performance, use the second analysis for a more realistic worst-case analysis.

### 35.16.5.4.1 Worst-case assumptions and formula

To estimate worst-case latency for a channel, assume this worst-case condition: the channel has just been serviced in a time slot of its priority level, and all other channels in the system are continuously requesting service and have cleared SGRs. The worst-case latency is the time from the end of the channel's service until the end of the channel's next service. See the following figure.

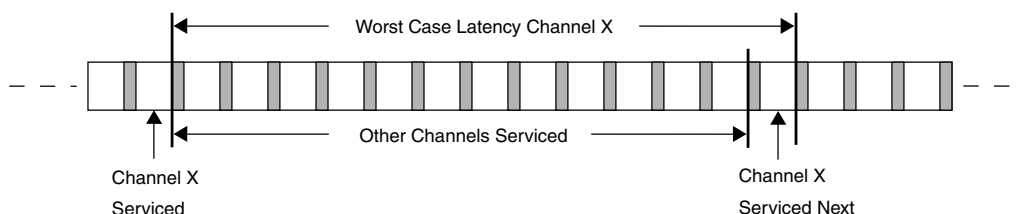


**Figure 35-339. First-pass Worst-Case Latency**

To estimate worst-case latency:

- Find the worst-case service time for each active channel.
- Using the H-M-H-L-H-M-H time-slot sequence, map the channels that are granted for each time slot.
- Add time for six-clock time-slot transitions.

#### 35.16.5.4.1.1 Finding the Worst-Case Service Time for Each Active Channel

A table for eTPU functions should list the longest threads (not counting initialization threads) for the functions, and the number of eTPU SDM accesses in the longest thread (semaphored and non semaphored). These figures will be used for estimating Microengine wait time. The table below is an example for old TPU functions in which there are only simple SDM accesses. It does not take into consideration the CDC operation and Microengine to Microengine communication.

The worst-case service time for each channel is: (CPCR=CCR=0)

Longest thread + ((number of RAM accesses in longest thread+1) \* RCR \* 2 clocks).

Note that the formula adds 1 RAM accesses for the parameter preload that occurs during TST. There are actually three accesses during TST, but only the first one can receive wait-states.

**Table 35-421. Longest Threads and RAM Accesses for old TPU Functions**

| Function | Longest Thread | RAM Accesses |
|----------|----------------|--------------|
| DIO      | 10             | 4            |

*Table continues on the next page...*

**Table 35-421. Longest Threads and RAM Accesses for old TPU Functions (continued)**

| Function         | Longest Thread               | RAM Accesses |
|------------------|------------------------------|--------------|
| ITC              | 40 (no linking) 42 (linking) | 7            |
| OC               | 40                           | 7            |
| PWM              | 24                           | 4            |
| SPWM             |                              |              |
| Mode 0           | 14                           | 4            |
| Mode1            | 18                           | 4            |
| Mode 2           | 20 (no linking)              | 4            |
|                  | 22 (linking)                 | 4            |
| PMA              | 94                           | 8            |
| PMM              | 94                           | 8            |
| PSP              |                              |              |
| Angle-Angle Mode | 76                           | 6            |
| Angle-Time Mode  | 50                           | 3            |
| SM <sup>1</sup>  | 160                          | 21           |
| PPWA             |                              |              |
| Mode 0           | 44                           | 9            |
| Mode 1           | 50 <sup>2</sup>              | 10           |
| Mode 2           | 44                           | 9            |
| Mode 3           | 50                           | 10           |

1. Assumes one master and one slave. For each additional slave
  1. Tab Add 32 clocks and 2 RAM accesses, and
  2. Tab Add (STEP\_RATE\_CNT \* two clocks)
2. With one channel linked. Add two clocks for each additional channel linked.

#### 35.16.5.4.1.2 Mapping the Channels for Each Time Slot

To determine when a channel will be serviced again, it is necessary to determine which other channels will be serviced first. Do this by assuming all channels are continuously requesting service and mapping the channels into the time-slot sequence.

#### 35.16.5.4.1.3 Adding Time for Time-Slot Transitions

Add six eTPU clocks for time-slot transitions which occur after each time slot.

#### 35.16.5.4.2 First-pass analysis Worst-Case Latency examples

The examples in this section assume the system configuration shown in the following table.

**Table 35-422. System configuration example**

| Channel | Priority | Function <sup>1, 2</sup>                    |
|---------|----------|---------------------------------------------|
| 0       | High     | PWM (driving a DC motor)                    |
| 1       | Middle   | PPWA (Mode 0, measuring the DC motor speed) |
| 2       | Low      | DIO (Input)                                 |

1. 9% RAM Collision Rate (RCR)

2. CPU clock rate = 40 MHz, or 25 ns per clock period

### 35.16.5.4.2.1 Finding the WCL for PWM on Channel 0

The following shows how to find the WCL for PWM on channel 0.

1. Find the worst-case service time for each active channel.

a. Longest thread of PWM is 24 CPU clocks with four RAM accesses.

$24 + ((4 \text{ RAM accesses} + 1) * 0.09 * 2 \text{ CPU clock waits}) = 24.9 \text{ CPU clocks}$ ,  
rounded up to 25 CPU clocks (since there are no partial clock periods)

Channel 0 worst-case service time = 25 CPU clocks.

b. Longest thread of PPWA in mode 0 is 44 CPU clocks with nine RAM accesses.

$44 + ((9 \text{ RAM accesses} + 1) * 0.09 * 2 \text{ CPU clock waits}) = 45.8 \text{ CPU clocks}$ ,  
rounded up to 46 CPU clocks

Channel 1 worst-case service time = 46 CPU clocks.

c. Longest thread of DIO is ten CPU clocks with four RAM accesses.

$10 + ((4 \text{ RAM accesses} + 1) * 0.09 * 2 \text{ CPU clock waits}) = 10.9 \text{ CPU clocks}$ ,  
rounded up to 11 CPU clocks

Channel 2 worst-case service time = 11 CPU clocks.

2. Assume channel 0 has just been serviced and that channels 1 and 2 are continuously requesting service. Using the H-M-H-L-H-M-H time-slot sequence, map the channels that are granted for each time slot. See the following figure.

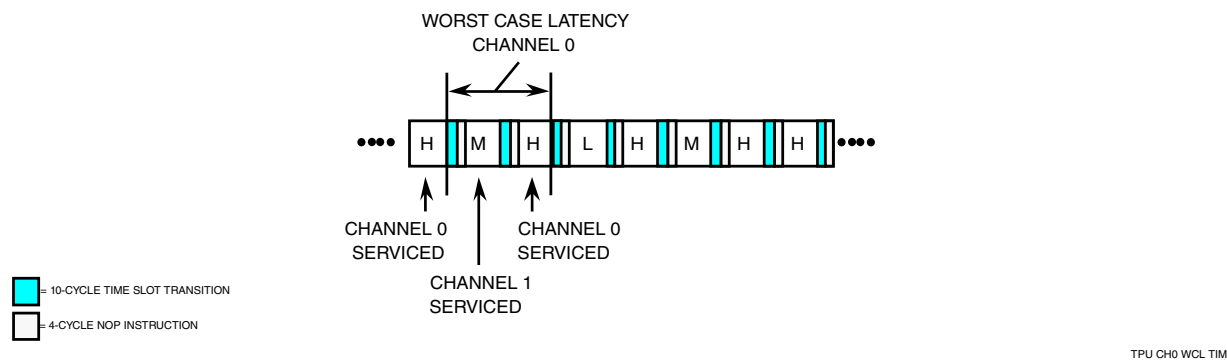


Figure 35-340. Next Servicing for Channel 0

Channel 1 will be serviced in the middle-priority time slot before channel 0 is serviced again.

3. Add time for the six-clock CPU time-slot transitions. See the above figure and the following table.

A four-clock NOP occurs after each channel is serviced since there is one channel in each priority level, i.e., a new cycle for a priority level is started after each channel is serviced. Time-slot transitions occur after each time slot.

Table 35-423. Worst-Case Latency for Channel 0

|                                   |           |
|-----------------------------------|-----------|
| Channel 0 worst-case service time | 25 clocks |
| Channel 1 worst-case service time | 46 clocks |
| Two 6-clock time-slot transitions | 12 clocks |
| Total clocks                      | 83 clocks |

$83 \text{ clocks} * 25 \text{ ns/clock} = 2075 \text{ ns}$

Conclusion: in this system configuration PWM can run with a minimum high time or low time of 2075 ns.

Note that in double match eTPU system the PWM can be serviced once in each period, and there is no latency for minimum high time. The latency in eTPU PWM function will represent the minimum PWM period.

35.16.5.4.2.2 Finding the WCL for PPWA on Channel 1

The following shows how to find the WCL for PPWA on channel 1.

1. Find the worst-case service time for each active channel. See step 1 of previous example.
2. Assume channel 1 has just been serviced and that channels 0 and 2 are continuously requesting service. Using the H-M-H-L-H-M-H time-slot sequence, map the channels that are granted for each time slot. See the following figure.

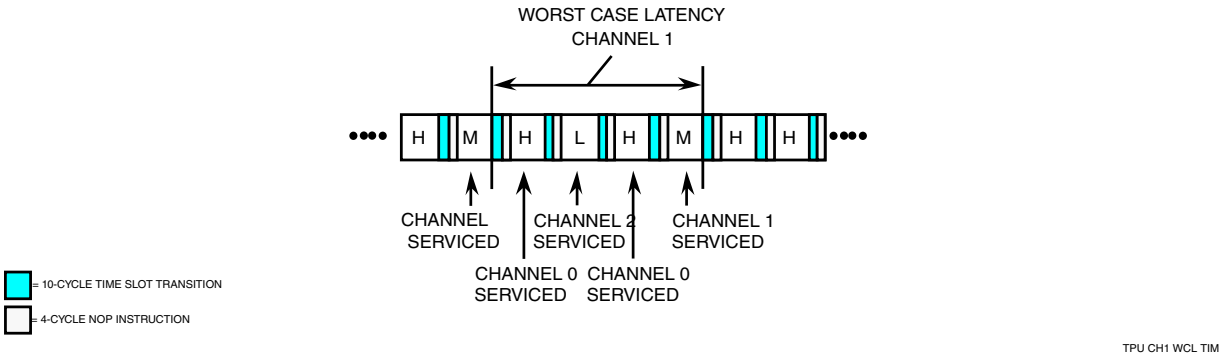


**Figure 35-341. Next Servicing for Channel 1**

Channel 0 will be serviced twice and channel 2 once before channel 1 is serviced again.

3. Add time for the six-clock CPU time-slot transitions. See the above figure and the following table.

**Table 35-424. Worst Case Latency for Channel 1**

|                                        |            |
|----------------------------------------|------------|
| Two Channel 0 worst-case service times | 50 clocks  |
| Channel 1 worst-case service time      | 46 clocks  |
| Channel 2 worst-case service time      | 11 clocks  |
| Four 6-clock time-slot transitions     | 24 clocks  |
| Total clocks                           | 131 clocks |

$$131 \text{ clocks} * 25 \text{ ns/clock} = 3275 \text{ ns}$$

Conclusion: in this system configuration PPWA can measure a period or pulse of minimum 3275 ns.

Note that PPWA function optimized for eTPU hardware can use double transition mode to measure very narrow pulses with one service after the second transition, and latency will affect only the minimum gap between two input pulses. Also the function threads would have more efficient coding.

35.16.5.4.2.3 Finding the WCL for DIO on Channel 2

The following shows how to find the WCL for DIO on channel 2.

- 1. Find the worst-case service time for each active channel. See step 1 of previous examples.
- 2. Assume channel 2 has just been serviced and that channels 0 and 1 are continuously requesting service. Using the H-M-H-L-H-M-H-H-M-H-L-H time-slot sequence, map the channels that are granted for each time slot. See the following figure.

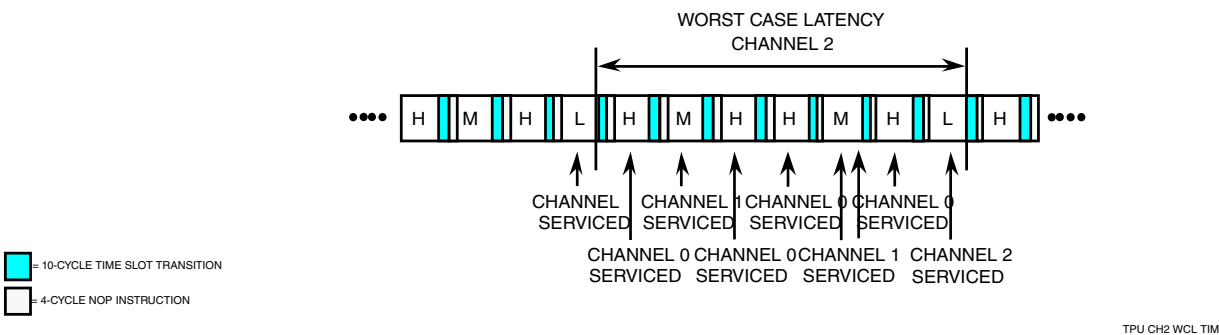


Figure 35-342. Next Servicing for Channel 2

Channel 0 will be serviced four times and channel 1 twice before channel 2 is serviced again.

- 3. Add time for the ten-clock CPU time-slot transitions and the four-clock NOPs. See the above figure and the following figure.

Table 35-425. Worst Case Latency for Channel 2

|                                         |            |
|-----------------------------------------|------------|
| Four Channel 0 worst-case service times | 100 clocks |
| Two Channel 1 worst-case service time   | 92 clocks  |
| Channel 2 worst-case service time       | 11 clocks  |
| Seven 6-clock time-slot transitions     | 42 clocks  |
| Total clocks                            | 245 clocks |

245 clocks \* 25 ns/clock = 6125 ns

Conclusion: in this system configuration DIO can keep track of the input level at a minimum of every 6125 ns.



Note that DIO function optimized for eTPU hardware can use double transition mode to measure two pin transitions at a time and reduce the service time, improving the overall system performance and latency.

### 35.16.5.5 Second-Pass Worst-Case Latency Analysis

Following is an example of a second-pass analysis for calculating worst-case latency for a channel. The second-pass analysis is useful for higher-performance systems, since it gives a more realistic worst-case latency result than first-pass analysis.

This example uses a relatively simple system in order to illustrate the basic principles of second-pass analysis. For a more complex example of second-pass analysis, refer to Multiphase Motor Commutation TPU Function (COMM)(TPUPN09/ D).

#### 35.16.5.5.1 Second-Pass Analysis Guidelines

Rather than use a fixed formula, a second-pass analysis relies on the application of the following guidelines.

1. The first-pass analysis makes the assumption that all channels in the system are continually requesting service. For many systems this is an unrealistic assumption. For example, if TCR1 is counting at a rate of 2 MHz (500 ns per count) and a channel is running the DIO function with a match rate of 20,000 TCR1 counts, the DIO will request service every 10 ms ( $20,000 * 500 \text{ ns} = 10,000,000 \text{ ns}$  or 10 ms). It is therefore unrealistic to assume that the channel running this DIO function is continuously requesting service. Figure out a realistic service request rate for each channel. Time slots can then be mapped to each channel at the real rate of request.
2. If a function is active during system initialization but not during the high-speed running mode of the system, then that system does not need to be included in the high-speed worst-case latency calculations.
3. Use a realistic SDM collision rate.
4. Be careful when assigning functions priority levels and channel numbers. Decide which function or functions will be most difficult to make perform at the desired level. Assign those channels high priority and low channel numbers. Try different priority and channel assignments to see how it affects the system.
5. The seven-slot sequence of || H | M | H | L | H | M | H || is asymmetrical when put back-to-back with other seven-slot sequences. Note that in the following sequence there are two high-priority slots next to each other:

|| H | M | H | L | H | M | H ||| H | M | H | L | H | M | H ||

6. Make sure that when mapping out channels to the sequence, you choose a worst-case slot to start the mapping. For example, when estimating WCL for a high-priority channel, do not start the mapping in the last high-priority slot in a seven-slot sequence, as that is a best case for a high-priority channel since another high-priority time slot is next.
7. Instead of always using the longest thread in the function as the worst-case thread, evaluate the threads in the function that will be used in the system and use the appropriate worst-case threads. For example, in the preceding example of first-pass analysis, the PWM was shown to be able to achieve a high time and low time of 2475 ns under worst-case conditions. This was derived using the longest PWM thread of 24 CPU clocks. This longest thread is actually thread 2, the thread that is entered after the pin has just gone high. Thread 3, the thread that is entered after the pin has just gone low, requires only 2 CPU clocks. Therefore, in the first-pass example, the high time was correctly derived, but the low time is actually shorter than was estimated.

### 35.16.5.5.2 Second-pass analysis example

This example requires three 50% PWM waveforms: one 5 kHz (200 ms/period) and two 50 kHz (20 ms/period), each running DC motors. (Remember that the PWM function requests service from the eTPU after each high time and after each low time, so the eTPU must handle a request every 100 ms for the 5 kHz PWM and every 10 ms for the 50 MHz PWM.)

#### Note

This example uses square waves for simplicity. Notice that to use a PWM waveform in the typical way, in which the pulse is modulated, the pulse must not be modulated in a way that violates the worst-case latency requirements.

This example also uses one DIO channel monitoring a signal level every millisecond and one PPWA channel in mode 0 monitoring the speed of the 5-kHz DC motor. The PPWA must measure periods of 5 kHz (200 ms/period).

The CPU is interrupted by the channel running the PPWA function after measuring 200 periods (every 40 ms). The interrupt service routine performs an averaging of the period accumulation and checks it against a known parameter. The interrupt service time is so short and infrequent that it is a tiny fraction of total system time. The interrupt service routine contains no polling of the SDM. Therefore a realistic RCR = 0%.

### 35.16.5.5.2.1 First-Try System Configuration

Try a system configuration that seems likely to work. If it does not, change priority levels or channel numbers.

The 5 kHz and 50 kHz PWMs are the most time-critical functions. Those are assigned high priority. PPWA is assigned middle priority. The DIO is low performance and is assigned low priority. Refer to the following table.

**Table 35-426. First-Try System Configuration**

| Channel | Priority | Function <sup>1, 2</sup>          |
|---------|----------|-----------------------------------|
| 0       | High     | PWM at 50 kHz (needs a 4-μs WCL)  |
| 1       | High     | PWM at 50 kHz (needs a 4-μs WCL)  |
| 2       | High     | PWM at 5 kHz (needs a 40-μs WCL)  |
| 8       | Middle   | PPWA at 5 kHz (needs a 80-μs WCL) |
| 15      | Low      | DIO as input at rate of 1 ms      |

1. 0% RAM collision rate

2. CPU clock rate = 40 MHz, or 60 ns per clock period

With this system configuration, worst-case service time for each active channel is determined as follows:

1. Longest thread of PWM is 24 CPU clocks with four RAM accesses.

$$24 + ((4 \text{ RAM accesses} + 1) * 0 * 2 \text{ CPU clock waits}) = 24 \text{ CPU clocks}$$

Channels 0-2 worst-case service time = 24 CPU clocks.

2. Longest thread of PPWA in mode 0 is 44 CPU clocks with nine RAM accesses.

$$44 + ((9 \text{ RAM accesses} + 1) * 0 * 2 \text{ CPU clock waits}) = 44 \text{ CPU clocks}$$

Channel 8 worst-case service time = 44 CPU clocks.

3. Longest thread of DIO is ten CPU clocks with four RAM accesses.

$$10 + ((4 \text{ RAM accesses} + 1) * 0 * 2 \text{ CPU clock waits}) = 10 \text{ CPU clocks}$$

Channel 15 worst-case service time = 10 CPU clocks.

To find the WCL for channel 0, assume channel 0 has just finished service.

Map the channels in the H-M-H-L-H-M-H sequence. See the following figure.

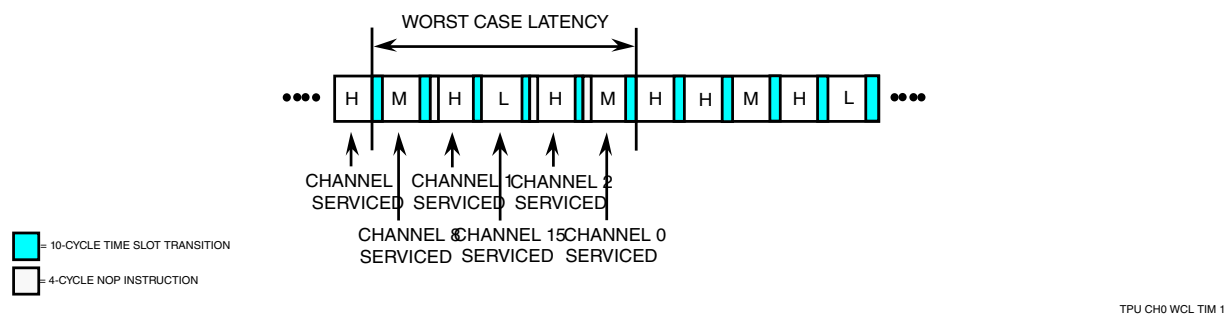


Figure 35-343. Worst-Case Latency for Channel 0 (First Try)

Conclusion: with this system configuration, worst-case latencies for channels 0 and 1 are too high (WCL for channel 1 is the same as WCL for channel 0). Try a different system configuration.

35.16.5.5.2.2 Second-Try System Configuration

The second-try system configuration is shown in the following table.

Table 35-427. Second-Try System Configuration

| Channel | Priority | Function <sup>1, 2</sup>          |
|---------|----------|-----------------------------------|
| 0       | High     | PWM at 50 kHz (needs a 4-μs WCL)  |
| 1       | High     | PWM at 50 kHz (needs a 4-μs WCL)  |
| 2       | Middle   | PWM at 5 kHz (needs a 40-μs WCL)  |
| 8       | Middle   | PPWA at 5 kHz (needs a 80-μs WCL) |
| 15      | Low      | DIO as input at rate of 1 ms      |

- 1. 0% RAM collision rate
- 2. CPU clock rate = 40 MHz, or 60 ns per clock period

To find the WCL for channel 0, assume channel 0 has just finished service. Map the channels in the H-M-H-L-H-M-H sequence. See the following figure.

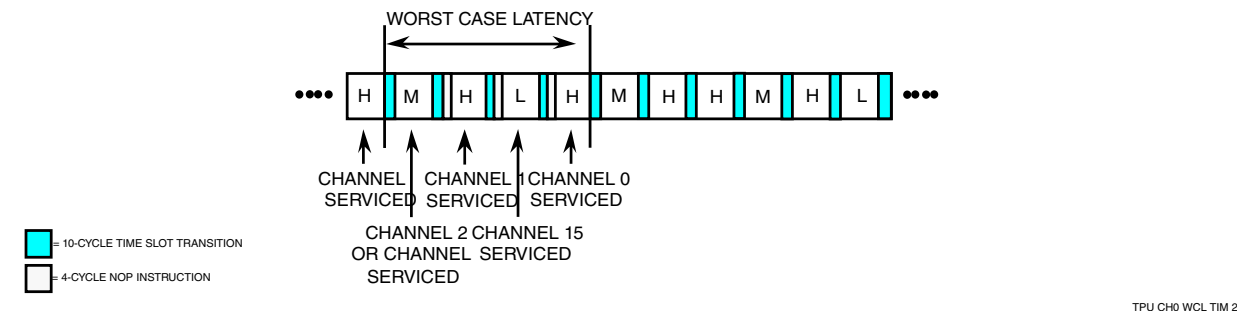


Figure 35-344. Worst-Case Latency for Channel 0 (Second Try)

Conclusion: with this system configuration, the WCL of both channel 0 and channel 1 is 3.85 ms, which is within the limit of 4 ms needed for a 50-kHz PWM.

Next, find the WCL for channel 2. Assume channel 2 has just finished service. Map the channels in the H-M-H-L-H-M-H sequence. See the following figure.

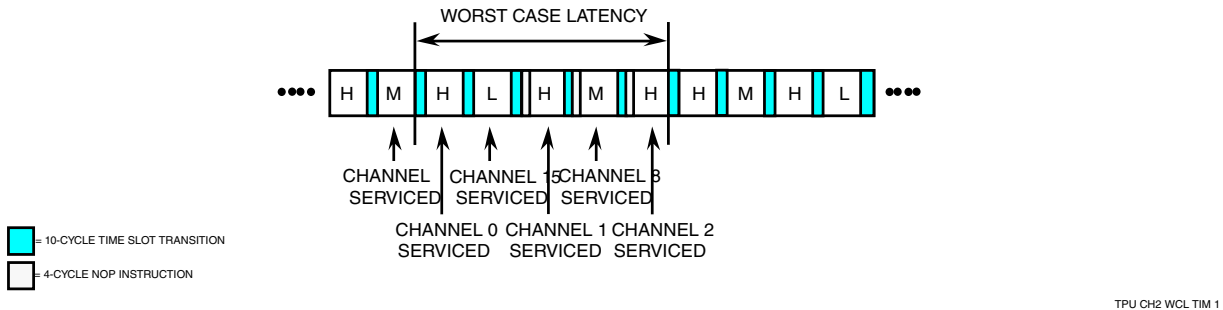


**Figure 35-345. Worst-Case Latency for Channel 2**

Conclusion: with this system configuration, the WCL for channels 2 and 8 is 4.7 ms, which is within the 40 and 80 ms WCL requirements.

Notice that channels 2 and 8 are well within their WCL requirements. The system could be reconfigured as shown in the following table to give channels 0 and 1 a larger margin while still keeping channels 2, 8 and 15 within their WCL requirements.

**Table 35-428. Second-Try System with Channel 0 and 1 Reconfigured**

| Channel | Priority | Function <sup>1, 2</sup>          |
|---------|----------|-----------------------------------|
| 0       | High     | PWM at 50 kHz (needs a 10-μs WCL) |
| 1       | High     | PWM at 50 kHz (needs a 10-μs WCL) |
| 2       | Middle   | PWM at 5 kHz (needs a 40-μs WCL)  |
| 8       | Low      | PPWA at 5 kHz (needs a 80-μs WCL) |
| 15      | Low      | DIO as input at rate of 1 ms      |

1. 0% RAM collision rate

2. CPU clock rate = 40 MHz, or 60 ns per clock period

## 35.16.6 Endianness

The following table shows the parameter addresses for big and little endian machines.

**Table 35-429. Parameter Addresses and Endianness**

| Parameter                                | byte address offset<br>(n = word address offset) |               |
|------------------------------------------|--------------------------------------------------|---------------|
|                                          | big endian                                       | little endian |
| 32-bit                                   | 4*n                                              |               |
| 24-bit                                   | 4*n + 1                                          | 4*n           |
| 32-bit parameter's most significant byte | 4*n                                              | 4*n + 3       |
| 24-bit parameter's most significant byte | 4*n + 1                                          | 4*n + 2       |
| least significant byte                   | 4*n + 3                                          | 4*n           |

### 35.16.7 Memory Error Service Procedures

Follows separate procedures for correctable and non-correctable error interrupt service routines. It is assumed that the non-correctable error ISR has higher priority, and cannot be preempted by the correctable error ISR. For memory error support details, see [Memory error support](#).

#### Note

The memory error support hardware does not (and is not intended to) guarantee that no error occurred will go unreported. However, combined with the procedures below, the information loss is minimized to statistical irrelevance on most scenarios.

#### 35.16.7.1 Non-correctable error service procedure

The basic procedure below can be followed in a Global Exception or a bus error machine check service routine. It handles the service for both SCM and SDM errors, as they are symmetrical at this level. However, a real implementation is likely to split the procedure in two, one for each memory.

1. Confirm/identify the error source, checking assertion of the [eTPU Module Configuration Register \(eTPU\\_SCR\\_MCR\)](#) fields SCMERR and SDMERR.
2. Save the state(s) of [eTPU Module Configuration Register \(eTPU\\_SCR\\_MCR\)](#) fields CRDIS/DRDIS.
3. Set the [eTPU Module Configuration Register \(eTPU\\_SCR\\_MCR\)](#) CRDIS/DRDIS=1 to disable further updates to the respective report registers, guaranteeing coherency.

4. Check whether the report registers were updated by a correctable or non-correctable error; to do this, check the [eTPU Code Error Report Syndrome Register \(eTPU\\_MESR\\_CERSR\)](#) syndrome (for SCM) and/or the [eTPU Data Error Report Syndrome Register \(eTPU\\_MESR\\_DERSR\)](#) syndromes (for SDM, only those with the respective BE flags asserted). If it is a correctable error report, skip the steps 5 and 6 for the respective memory.
5. Identify the error source (host, microengine or MISC/CDC) by checking the field ERR\_ACC in the register(s) [eTPU Code Error Report Address Register \(eTPU\\_MESR\\_CERAR\)](#) and [eTPU Data Error Report Address Register \(eTPU\\_MESR\\_DERAR\)](#); take or schedule the appropriate application-dependent recover actions.
6. Update error statistics as needed; it may include recording the access source, address, data and syndrome from:
  - [eTPU Code Error Report Address Register \(eTPU\\_MESR\\_CERAR\)](#)
  - [eTPU Data Error Report Address Register \(eTPU\\_MESR\\_DERAR\)](#)
  - [eTPU Code Error Report Data Register \(eTPU\\_MESR\\_CERDR\)](#)
  - [eTPU Data Error Report Data Register \(eTPU\\_MESR\\_DERDR\)](#)
  - [eTPU Code Error Report Syndrome Register \(eTPU\\_MESR\\_CERSR\)](#)
  - [eTPU Data Error Report Syndrome Register \(eTPU\\_MESR\\_DERSR\)](#)
7. Save the status of non-correctable error overflows flagged in the [eTPU Memory Error Status Register \(eTPU\\_MESR\\_MESR\)](#) fields CNCOVR/DNCOVR.
8. Set the [eTPU Memory Error Control Register \(eTPU\\_MESR\\_MECR\)](#) fields CRDIS/DRDIS to the state(s) saved in the step 2. That assures the access coherency of a preempted Correctable Error Service Procedure (which may have set CRDIS/DRDIS before this procedure) is not disturbed.
9. Clear the [eTPU Memory Error Status Register \(eTPU\\_MESR\\_MESR\)](#) fields CNCOVR/DNCOVR and CNCERR/DNCERR at once, in a single write to eTPU\_MESR\_MESR.
10. Use the overrun status saved on step 7 to update the error statistics, as needed.
11. Check the other Global Exception flags in [eTPU Module Configuration Register \(eTPU\\_SCR\\_MCR\)](#); take or schedule the appropriate (application dependent) recover actions.
12. Clear the Global Exception, programming [GEC](#) =1.

### 35.16.7.2 Correctable error service procedure

The basic procedure below can be followed in a Memory Error Interrupt service routine (see [Interrupts and data transfer requests](#)). It handles the service for both SCM and SDM errors, as they are symmetrical at this level. However, a real implementation is likely to split the procedure in two, one for each memory.

1. Confirm/identify the error source, checking assertion of the [eTPU Memory Error Status Register \(eTPU\\_MESR\\_MESR\)](#) fields CCERR/DCERR.
2. Program the [eTPU Memory Error Control Register \(eTPU\\_MESR\\_MECR\)](#) fields CRDIS=1 and/or DRDIS=1, to disable further updates to the respective report registers, guaranteeing coherency.
3. Check whether the report registers were updated by a correctable or non-correctable error; to do this, check the [eTPU Code Error Report Syndrome Register \(eTPU\\_MESR\\_CERSR\)](#) syndrome (for SCM) and/or the [eTPU Data Error Report Syndrome Register \(eTPU\\_MESR\\_DERSR\)](#) syndromes (for SDM, only those with the respective BE flags asserted). If it is a non-correctable error report, skip the steps 4 and 5 for the respective memory.
4. Identify the error source (host, microengine or MISC/CDC) by checking the field ERR\_ACC in [eTPU Code Error Report Address Register \(eTPU\\_MESR\\_CERAR\)](#) and [eTPU Data Error Report Address Register \(eTPU\\_MESR\\_DERAR\)](#); take or schedule the appropriate application-dependent recover actions, which may include soft-error fixing (see [Error fixing](#)).
5. Update error statistics as needed; it may include recording the access source, address, data and syndrome from:
  - [eTPU Code Error Report Address Register \(eTPU\\_MESR\\_CERAR\)](#)
  - [eTPU Data Error Report Address Register \(eTPU\\_MESR\\_DERAR\)](#)
  - [eTPU Code Error Report Data Register \(eTPU\\_MESR\\_CERDR\)](#)
  - [eTPU Data Error Report Data Register \(eTPU\\_MESR\\_DERDR\)](#)
  - [eTPU Code Error Report Syndrome Register \(eTPU\\_MESR\\_CERSR\)](#)
  - [eTPU Data Error Report Syndrome Register \(eTPU\\_MESR\\_DERSR\)](#)
6. Save the status of correctable error overflows flagged in the [eTPU Memory Error Status Register \(eTPU\\_MESR\\_MESR\)](#) fields CCOVR/DCOVR.
7. Program the [eTPU Memory Error Control Register \(eTPU\\_MESR\\_MECR\)](#) fields CRDIS/DRDIS=0.
8. Clear the [eTPU Memory Error Status Register \(eTPU\\_MESR\\_MESR\)](#) fields CCOVR/DCOVR and CCERR/DCERR at once, in a single write to [eTPU\\_MESR\\_MESR](#).



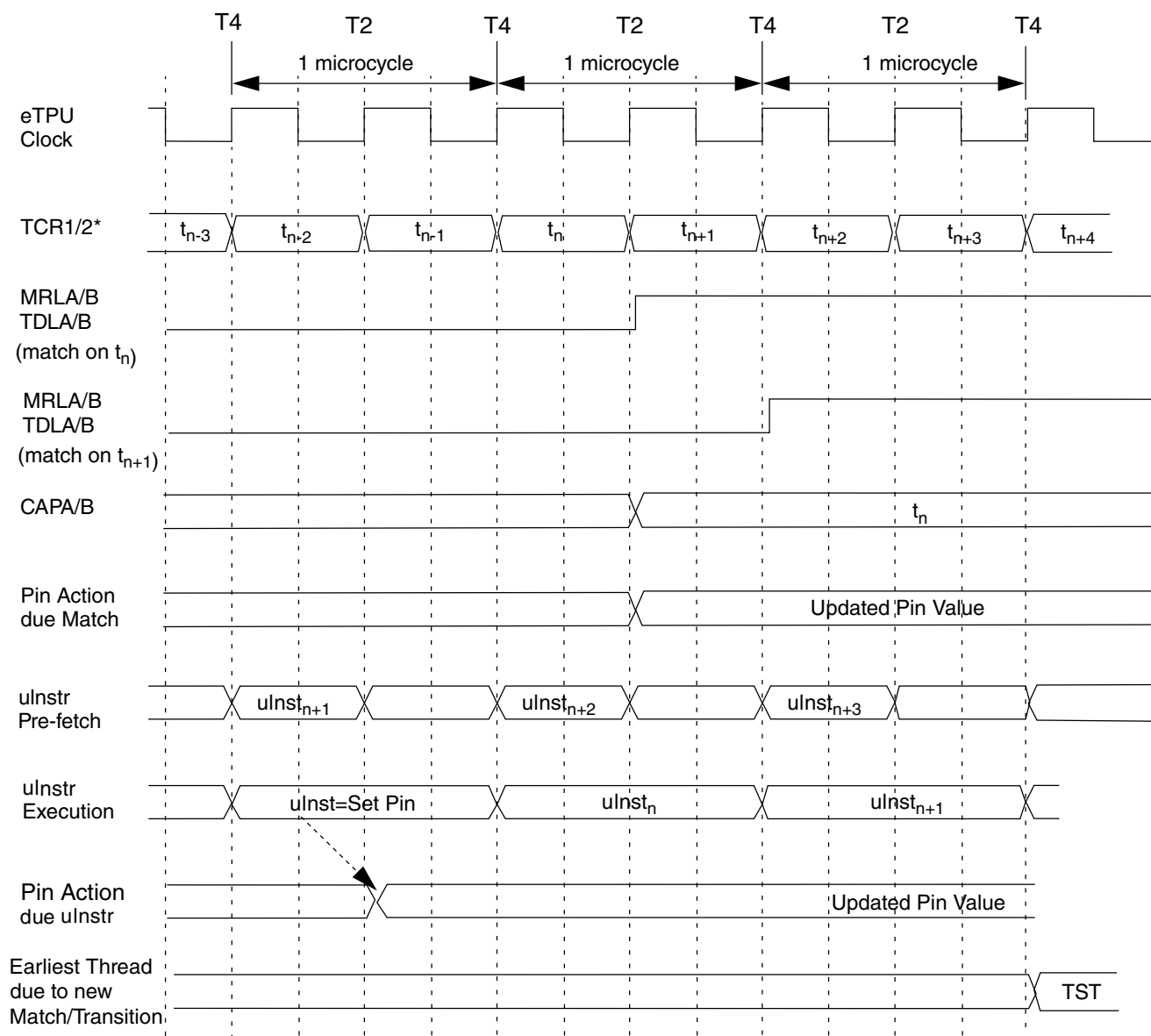
9. Use the saved overrun status to update the error statistics, as needed.

## **35.17 Appendices**

### **35.17.1 Microcycle and I/O timing**

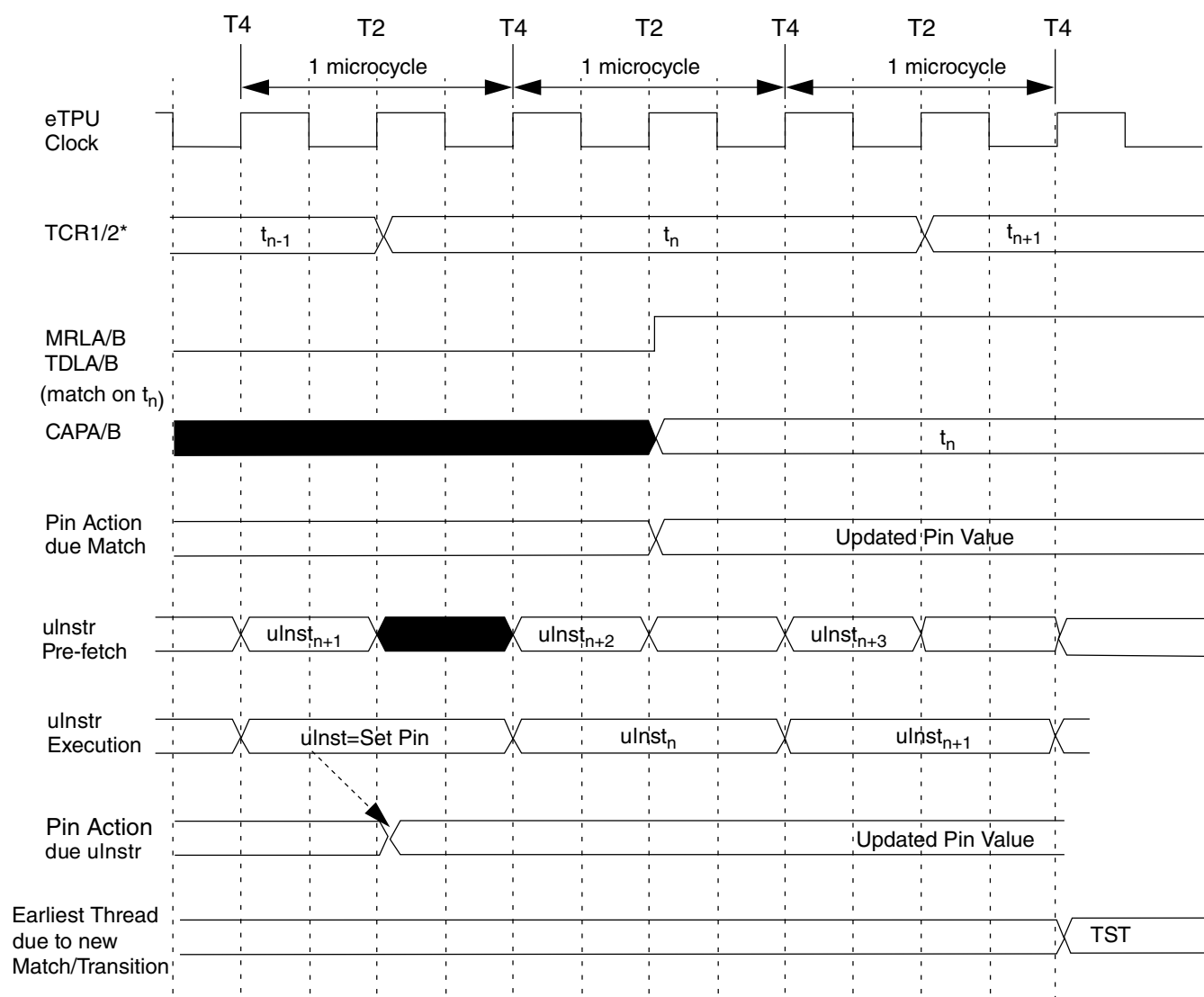
#### **35.17.1.1 Execution and channel timing**

The following figure shows the main timings related to microinstruction execution when channels and timebases run on T2 timing.



Note: \*TCR clock/prescaler selection = 1 x eTPU clock

**Figure 35-346. Execution, timebase and channel T2 timing**



**Note:** \*TCR clock/prescaler selection = 4x eTPU clock

**Figure 35-347. Execution, timebase and channel T2/T4 timing**

The sequential occurrence of the four T states (T1-T4) constitutes a microcycle. Only T2 and T4 are taken as reference for timings, either internal or external. T2 and T4 have the timing of the positive eTPU clock pulses, and are used in most of the eTPU logic in an edge-triggered design style.

Two additional T states are derived from the eTPU clocks:  $\overline{T2}$  and  $\overline{T4}$ .  $\overline{T2}$  occurs when the eTPU loses SDM arbitration to a bus master.  $\overline{T4}$  occurs in halt state (due to a breakpoint or device debug request assertion, for instance), or in stall state (due to an NDEDI queue full); see [Development support features](#) for more details.

$\overline{T2}$  and  $\overline{T4}$  states are defined as microcycle timing states (not to be confounded with logic states) of one eTPU clock in which the T clocks continue to run, but the control signals associated with the clocks are unaffected. That is, no operation occurs during these states.

Both  $\overline{T2}$  and  $\overline{T4}$  states occur in multiples of two eTPU clocks to keep the microengine synchronized with the free running channels, which are unaffected and keep on working as in T2 and T4.

Thus, the eTPU has two types of timing states:

- $\overline{T2}$ : hold execution for SDM access, from clock pulse T2 until one of the next T2 clock pulses of another microcycle.
- $\overline{T4}$ : hold execution in debug mode or stall, from clock pulse T4 until one of the next T4 clock pulses of another microcycle.

The two following figures show the timing of  $\overline{T2}$  and  $\overline{T4}$  timing states, respectively.

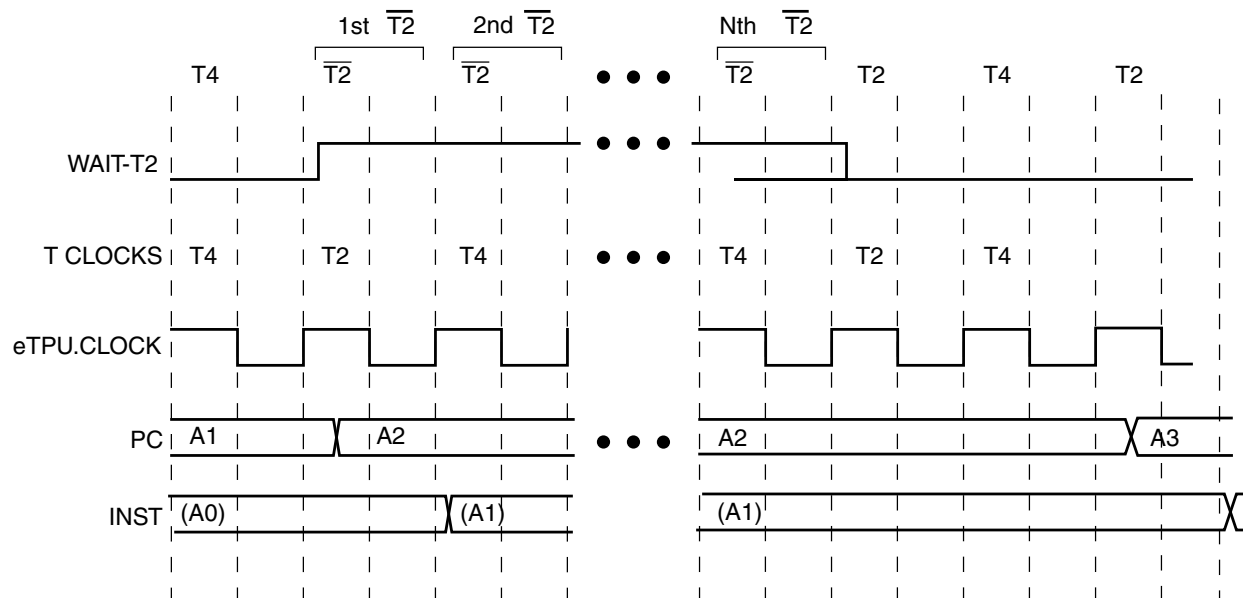
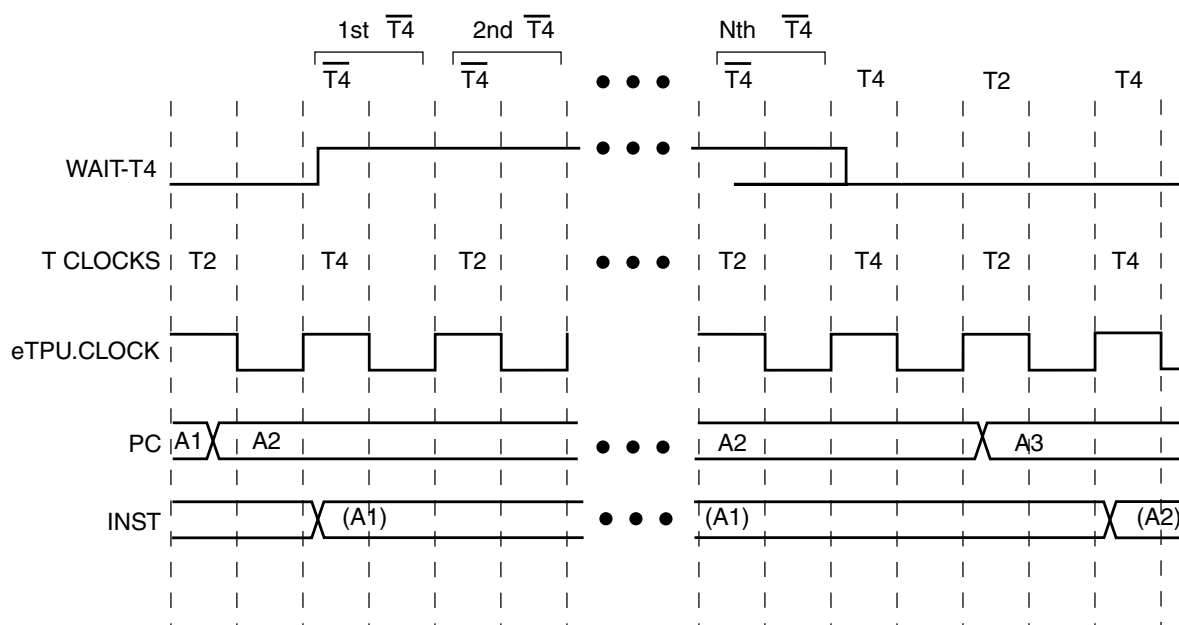


Figure 35-348.  $\overline{T2}$  timing

Figure 35-349.  $\overline{T4}$  timing

### 35.17.1.2 Input/Output Signal Delays

The synchronizer, filter and edge detection logic delay the input signal transitions.

The Filter Delay varies with the filter clock ([eTPU Engine 1 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG1\)](#) and [eTPU Engine 2 Configuration Register \(eTPU\\_SCR\\_ECR\\_ENG2\)](#) field FPSCK) and the filter mode used, as shown in the [Table 35-355](#). For any given transition, it depends on the phase of the filter clock when the input transition happens. In integration mode (TCRCLK filtering only), it also depends on the state of the integrator counter. The Total Delay is defined as the number of eTPU clock rising edges between the input transition and the setting of TDLA/B, TCR1/2 incrementing, or EAC tooth sensing (TCRCLK) in angle mode. The synchronizer delay is 2 or 3 eTPU clocks, depending on the phase of the synchronizer when the input transition happens. The edge detection takes 1 more eTPU clock. The total delays are, thus:

```
Min. Total Delay = Min. Synchronizer Delay + Min. Filter Delay + Edge Detection Delay
Min. Total Delay = 3 + Min. Filter Delay
Max. Total Delay = Max. Synchronizer Delay + Max. Filter Delay + Edge Detection Delay
Max. Total Delay = 4 + Max. Filter Delay
```

The channel filters can be bypassed, so nullifying the filter delays in the equations above.

See the chip-specific eTPU information for connections between eTPU and input glitch filters.

The channel output flip-flops drive the eTPU output signals directly, without any synchronous delays.

## 35.17.2 Channel Mode Summary

The initial state considered for all submodes is channel flags MRLEA and MRLEB asserted, MRLA, MRLB, TDLA and TDLB negated. From initial state one can follow the table below and verify how each submode behaves in a determined sequence of events. Note that the actions performed by an event type depend on all previous events following the initial state, for a given channel submode.

There are three columns for each event: one for event type, one for enable/disable actions and one for capture. **Event type** column can be matchA, matchB, transA and transB (for double transition modes). **Enable/disable actions** column (identified as "[blocks] (enables)" in column head) specifies which other events are enabled or disabled. Initially disabled events (specified in "initially blocked" column) are usually enabled by other events.

In double transition submodes, the first transition detected is always considered transA and the second is considered transB. This means that transA event actually enables the detection of transB event. This is not explicit in the table, since it is a general behavior for all double transition submodes.

A sequence of four events (two matches and two transitions) are necessary to describe the behavior of some channel submodes. When a determined sequence of events has less than four events, the other event columns are left blank.

Cells in an "event type" column that have light-grayed background indicate that a service request is generated. More than one event in the same event sequence can issue service request.

### Note

The two tables below do not exhaust all possibilities of channel logic event sequences, because it does not account for possible microcode interventions. For instance, if matches are blocked by first transition and microcode resets TDLA, the matches become enabled again, and from this point on the channel behaves as if the first transition had never occurred.

**Table 35-430. Channel Mode Summary 1 of 2**

| Mode     | Initially Blocked | 1st Event              |                                 |         | 2nd Event  |                    |         |
|----------|-------------------|------------------------|---------------------------------|---------|------------|--------------------|---------|
|          |                   | Event Type             | [Blocks] (Enables) <sup>1</sup> | Capture | Event Type | [Blocks] (Enables) | Capture |
| em_nb_st | none              | match 1/2 <sup>2</sup> | none                            | 1/2     | match 2/1  | none               | 2/1     |

*Table continues on the next page...*

**Table 35-430. Channel Mode Summary 1 of 2 (continued)**

| Mode               | Initially Blocked | 1st Event           |                                 |         | 2nd Event  |                    |         |
|--------------------|-------------------|---------------------|---------------------------------|---------|------------|--------------------|---------|
|                    |                   | Event Type          | [Blocks] (Enables) <sup>1</sup> | Capture | Event Type | [Blocks] (Enables) | Capture |
| em_nb_dt           | none              |                     |                                 |         | trans 1    | [matches]          | both    |
|                    |                   | trans 1             | [matches]                       | both    | trans 2    | none               | 2       |
|                    |                   | match 1/2           | none                            | 1/2     | match 2/1  | none               | 2/1     |
|                    |                   | match 1             | none                            | 1       | trans 1    | none               | 1       |
|                    |                   | match 2             | none                            | 2       | trans 1    | [match 1]          | 1       |
|                    |                   | trans 1             | [match 1]                       | 1       | match 2    | none               | 2       |
| em_b_st            | none              |                     |                                 |         | trans 2    | [match 2]          | 2       |
|                    |                   | match 1             | [match 2]                       | both    | trans 1    | [matches]          | both    |
|                    |                   | match 2             | [match 1]                       | both    |            |                    |         |
| em_b_dt            | none              | trans 1             | [matches]                       | both    | trans 2    | none               | 2       |
|                    |                   | match 1             | [match 2]                       | both    | trans 1    | none               | 1       |
|                    |                   | match 2             | [match 1]                       | both    |            |                    |         |
| bm_st              | none              | trans 1             | [match 1]                       | 1       | match 2    | none               | 2       |
|                    |                   | match 1/2           | none                            | 1/2     | trans 2    | [match 2]          | 2       |
|                    |                   | match 2/1           | none                            | 2/1     |            |                    |         |
| bm_dt              | none              | trans 1             | [matches]                       | both    | trans 1    | [matches]          | both    |
|                    |                   | trans 2             | none                            | 2       | trans 1    | none               | 1       |
|                    |                   | match 1/2           | none                            | none/2  |            |                    |         |
| m2_st              | trans 1           | trans 2             | [matches]                       | 2       | match 2    | none               | 2       |
|                    |                   | match 1             | (trans 1)                       | 1       | trans 1    | [matches]          | both    |
|                    |                   | match 1 and match 2 | (trans 1)                       | both    | trans 1    | [matches]          | both    |
| m2_dt              | trans 1           | match 2             | [match 1]                       | 2       | n/a        | n/a                | n/a     |
|                    |                   | match 1             | (trans 1)                       | 1       | trans 1    | none               | 1       |
|                    |                   | match 2             | (trans 1)                       | both    | match 2    | none               | 2       |
| m2_o_st            | trans 1 match 2   | match 1 and match2  | (trans 1)                       | both    | trans 1    | none               | 1       |
|                    |                   | match 2             | [match 1]                       | 2       | n/a        | n/a                | n/a     |
|                    |                   | match 1             | (match 2) (trans 1)             | 1       | trans 1    | [matches]          | both    |
| m2_o_dt            | trans 1 match 2   | match 1             | (match 2) (trans 1)             | 1       | match 2    | [trans 1]          | 2       |
|                    |                   | match 1             | (match 2) (trans 1)             | 1       | trans 1    | none               | 1       |
|                    |                   | match 2             | [trans 1]                       | 2       |            |                    |         |
| sm_st <sup>3</sup> | match 2           | trans 1             | none                            | both    | trans 1    | none               | both    |
|                    |                   | match 1             | [match 1]                       | both    | trans 2    | none               | 2       |
| sm_dt              | match 2           | match 1             | none                            | both    | trans 1    | none               | 1       |

Table continues on the next page...

**Table 35-430. Channel Mode Summary 1 of 2 (continued)**

| Mode                 | Initially Blocked | 1st Event  |                                 |         | 2nd Event  |                    |         |
|----------------------|-------------------|------------|---------------------------------|---------|------------|--------------------|---------|
|                      |                   | Event Type | [Blocks] (Enables) <sup>1</sup> | Capture | Event Type | [Blocks] (Enables) | Capture |
| sm_st_e <sup>4</sup> | match 2 trans 2   | trans 1    | none                            | 1       | match 1    | none               | 2       |
|                      |                   | trans 1    | none                            | 1       | trans 2    | [match1]           | 2       |
|                      |                   | match 1    | none                            | 1       | trans 1    | none               | 1       |
|                      |                   | trans 1    | [match 1]                       | 1       | n/a        | n/a                | n/a     |

1. Transition 1 always enables Transition 2.
2. Shading indicates that a service request is generated.
3. sm\_st is compatible with TPU3 channel logic.
4. It is not possible to include all functionality of this submode in table.

**Table 35-431. Channel Mode Summary 2 of 2**

| Mode     | Initially Blocked | 3rd Event            |                                 |          | 4th Event  |                    |         |
|----------|-------------------|----------------------|---------------------------------|----------|------------|--------------------|---------|
|          |                   | Event Type           | [Blocks] (Enables) <sup>1</sup> | Capture  | Event Type | [Blocks] (Enables) | Capture |
| em_nb_st | none              | trans 1 <sup>2</sup> | [matches]                       | both     | trans 2    | none               | 2       |
|          |                   |                      |                                 |          | n/a        | n/a                | n/a     |
|          |                   | trans 2              | none                            | 2        | trans 2    | none               | 2       |
| em_nb_dt | none              | trans 1              | none                            | 1        | trans 2    | none               | 2       |
|          |                   | match 2              | none                            | 2        | trans 2    | none               | 2       |
|          |                   | trans 2              | [match 2]                       | 2        | n/a        | n/a                | n/a     |
|          |                   | trans 2              | [match 2]                       | 2        | n/a        | n/a                | n/a     |
|          |                   | trans 2              | none                            | 2        | n/a        | n/a                | n/a     |
| em_b_st  | none              | trans 2              | none                            | 2        | n/a        | n/a                | n/a     |
| em_b_dt  | none              | trans 2              | none                            | 2        | n/a        | n/a                | n/a     |
|          |                   | trans 2              | none                            | 2        | n/a        | n/a                | n/a     |
| bm_st    | none              | trans 1              | [matches]                       | both     | trans 2    | none               | 2       |
|          |                   | trans 2              | none                            | 2        | n/a        | n/a                | n/a     |
| bm_dt    | none              | trans 1              | none                            | 1        | trans 2    | [matches]          | 2       |
|          |                   | match 2/1            | none                            | 2 / none | trans 2    | [matches]          | 2       |
|          |                   | trans 2              | [matches]                       | 2        | n/a        | n/a                | n/a     |
|          |                   | match 2/1            | none                            | 2 / none | trans 2    | [matches]          | 2       |
|          |                   | trans 2              | [matches]                       | 2        | n/a        | n/a                | n/a     |
| m2_st    | trans 1           | trans 1              | [matches]                       | both     | trans 2    | none               | 2       |
|          |                   | trans 2              | none                            | 2        | n/a        | n/a                | n/a     |
|          |                   | trans 2              | none                            | 2        | n/a        | n/a                | n/a     |
| m2_dt    | trans 1           | trans 2              | [match 2]                       | 2        | n/a        | n/a                | n/a     |
|          |                   | match 2              | none                            | 2        | trans 2    | none               | 2       |
|          |                   | trans 1              | none                            | 1        | trans 2    | none               | 2       |
|          |                   | trans 2              | none                            | 2        | n/a        | n/a                | n/a     |

Table continues on the next page...



**Table 35-431. Channel Mode Summary 2 of 2 (continued)**

| Mode                 | Initially Blocked | 3rd Event  |                                 |         | 4th Event  |                    |         |
|----------------------|-------------------|------------|---------------------------------|---------|------------|--------------------|---------|
|                      |                   | Event Type | [Blocks] (Enables) <sup>1</sup> | Capture | Event Type | [Blocks] (Enables) | Capture |
| m2_o_st              | trans 1 match 2   | trans 2    | none                            | 2       | n/a        | n/a                | n/a     |
| m2_o_dt              | trans 1 match 2   | trans 2    | [match 2]                       | 2       | n/a        | n/a                | n/a     |
|                      |                   | match 2    | [trans 2]                       | 2       | n/a        | n/a                | n/a     |
| sm_st <sup>3</sup>   | match 2           | trans 2    | none                            | 2       | n/a        | n/a                | n/a     |
| sm_dt                | match 2           | trans 2    | none                            | 2       | n/a        | n/a                | n/a     |
|                      |                   | trans 2    | none                            | 2       | n/a        | n/a                | n/a     |
| sm_st_e <sup>4</sup> | match 2 trans 2   | n/a        | n/a                             | n/a     | n/a        | n/a                | n/a     |

1. Transition 1 always enables Transition 2.
2. Shading indicates that a service request is generated.
3. sm\_st is compatible with TPU3 channel logic.
4. It is not possible to include all functionality of this submode in table.

### 35.17.3 MISC Algorithm

The MISC generator is based on the following polynomial:

$$G(x) = 1 + x^1 + x^2 + x^{22} + x^{31} \text{ (equivalent to feedback mask = 0x80400007)}$$

The MISC signature generation starts by clearing the MISC Accumulator value to 0 and preloading the MISC Counter with the highest SCM address. It then steps through each address decrementing the counter, reading 32 bit values and following the algorithm below:

```

If the least significant bit in MISC is 1 then
    MISC = MISC right shifted by 1 bit
    MISC = MISC XOR 0x80400007
else
    MISC = MISC right shifted by 1 bit
end if
MISC = MISC XOR RAM data

```

The code example below shows an excerpt of C code that calculates the MISC signature for a given array of data, based on the previous algorithm:

```

#define SCM_size    (MAX_SCM_ADDRESS / 4)    /* last byte address - converted to 32-bit word */
#define POLY        0x80400007               /* G(x) = 1 + x1 + x2 + x22 + x31 */
/*****
FUNCTION : void calc_misc()
PURPOSE : This function calculates the MISC value.
INPUTS NOTES : none
RETURNS NOTES : MISC value
GENERAL NOTES : the array 'unsigned int data[]' represents the actual memory
array, organized in 32-bit words.
*****/
unsigned int calc_misc (void)

```

```

{
    int j; /* loop counter */
    unsigned int misc = 0;
    for (j = (SCM_size-1); j >= 0 ; j--) { /* SCM_size has the number of 32-bit words in SCM
*/
        if (misc & 0x1) {
            misc >>= 1;
            misc ^= POLY;
        }
        else {
            misc >>= 1;
        }
        misc ^= data[j]; /* data[j] is the actual 32-bit word taken from the SCM
array
*/
    }
    return (misc); /* final signature calculated */
};

```

The value calculated by this algorithm must be loaded into [eTPU MISC Compare Register \(eTPU\\_SCR\\_MISCCMPR\)](#) prior to activating the SCM MISC calculator in eTPU. Once the MISC calculator is activated (bit SCMMISEN in register [eTPU Module Configuration Register \(eTPU\\_SCR\\_MCR\)](#) is written to 1) eTPU itself will start this procedure<sup>42</sup> reading the SCM whenever allowed by microengine. At the end of the cycle, when all the array has been read and the SCM signature is calculated, the Host CPU can be notified via Global Exception if the MISC Accumulator does not match the value in eTPU\_SCR\_MISCCMPR.

The average time taken by MISC to complete the signature of the whole SCM can be given by the formula:

$$\text{Average MISC period} = S / (4 * F * (1 - L))$$

where f is clock frequency, S is SCM size in bytes and L is eTPU load (as a percentage of execution clocks over a period of time, including TST clocks).

Further detail on MISC calculation can be found on [SCM Test - Multiple Input Signature Calculator](#). The application note **AN2192 - Detecting Errors in the Dual Port RAM (DPTRAM) Module** is also a good source of information (although it refers to TPU) on MISC signature.

42. eTPU MISC hardware is optimized to read 32-bit words from memory and to calculate this CRC in parallel, rather than shifting one bit at a time. The actual implementation inside eTPU, although bringing to the same results, does not match exactly the algorithm shown here.

## Chapter 36

# Input Glitch Filter (IGF)

### 36.1 Chip-specific IGF information

This chip has two instances of the IGF module: IGF\_0 and IGF\_1. They are in the eTPU clock domain. The two instances are located on different Peripheral Bridges.

#### 36.1.1 IGF channels and internal prescalers

The two IGF module instances support a combined total of 76 channels. Some channels of the instances implement internal prescalers.

Each channel has a 24-bit counter. Each internal prescaler is 9 bits wide.

The following table identifies the number of channels of each instance, the specific channel numbers that implement an internal prescaler, and the group of specific channel numbers that can use each implemented internal prescaler.

Each group of channel numbers that is associated with one implemented prescaler is a filter cluster.

**Table 36-1. IGF instance channels and internal prescalers**

| Instance | Channels supported | Channel number with prescaler | Filter cluster: channel numbers that use prescaler |
|----------|--------------------|-------------------------------|----------------------------------------------------|
| IGF_0    | 30                 | 0                             | 0-29                                               |
| IGF_1    | 46                 | 0                             | 0-31                                               |
|          |                    | 32                            | 32-45                                              |

### 36.1.2 IGF internal prescaler selection

Channels in a filter cluster that do not implement their own internal prescaler must select an "external" prescaler. This selection means the channel uses the prescaler of the preceding channel in the filter cluster.

Within a filter cluster, to enable a channel  $x$  to use the cluster's internal prescaler:

1. For the channel that implements the cluster's internal prescaler, select the internal prescaler by ensuring that the channel's  $MCR_n[PSSEL]$  field is 0.
2. For every channel between that channel and channel  $x$ , including channel  $x$  itself, select the "external" prescaler by setting the channel's  $MCR_n[PSSEL]$  field to 1.

#### 36.1.2.1 Example of internal prescaler selection

To select the applicable internal prescaler for IGF\_0 channel 3:

1. Because IGF\_0's channel 0 implements the cluster's internal prescaler, select the internal prescaler for channel 0 by ensuring that IGF\_0's  $MCR_0[PSSEL]$  field is 0.
2. Select the "external" prescaler for every channel between channel 0 and channel 3, including channel 3 itself:
  - a. Set IGF\_0's  $MCR_1[PSSEL]$  to 1.
  - b. Set IGF\_0's  $MCR_2[PSSEL]$  to 1.
  - c. Set IGF\_0's  $MCR_3[PSSEL]$  to 1.

#### 36.1.2.2 Restriction on prescaler selection for IGF\_1 channel 32

Do not set IGF\_1's  $MCR_{32}[PSSEL]$  field to 1.

This setting could cause timing failure for all IGF\_1 channels.

### 36.1.3 IGF interaction with other modules

The IGF modules interface with eTPU and eMIOS modules.

The following table shows the mapping of IGF\_0 channels.

**Table 36-2. IGF\_0 channel mapping**

| Filter channel | Primary function pin    | Module  |
|----------------|-------------------------|---------|
| 0              | TCRCLKC                 | eTPU_C  |
| 1-15           | ETPUC0-14               | eTPU_C  |
| 16-19          | EMIOS0_0-3 <sup>1</sup> | eMIOS_0 |
| 20-21          | EMIOS0_6-7              | eMIOS_0 |
| 22-29          | EMIOS0_16-23            | eMIOS_0 |

1. EMIOS0\_4 and EMIOS0\_5 do not have filters because the input channels are not connected directly from pads. They are connected from the DSPI\_B module.

The following table shows the mapping of IGF\_1 channels.

**Table 36-3. IGF\_1 channel mapping**

| Filter channel | Primary function pin    | Module  |
|----------------|-------------------------|---------|
| 0              | TCRCLKA                 | eTPU_A  |
| 1-15           | ETPUA0-14               | eTPU_A  |
| 16             | TCRCLKB                 | eTPU_B  |
| 17-31          | ETPUB0-14               | eTPU_B  |
| 32-37          | EMIOS1_0-5 <sup>1</sup> | eMIOS_1 |
| 38-45          | EMIOS1_16-23            | eMIOS_1 |

1. EMIOS1\_6 and EMIOS1\_7 do not have filters because the input channels are not connected directly from pads. They are connected from the DSPI\_D module.

### 36.1.4 STOP mode system requests for IGF\_0 and IGF\_1

As described in [STOP mode](#), a STOP mode system request can stop the IGF module instances.

This chip also has inter-module STOP mode system requests that affect IGF\_0 and IGF\_1 individually:

- IGF\_0 automatically stops when eTPU\_C and eMIOS\_0 stop.
- IGF\_1 automatically stops when eTPU\_A/eTPU\_B and eMIOS\_1 stop.

## 36.2 Introduction

The Input Glitch Filter provides programmable glitch rejection filtering of digital input signals.

### 36.3 Block Diagram

The figure below is the block diagram for the Input Glitch Filter module.

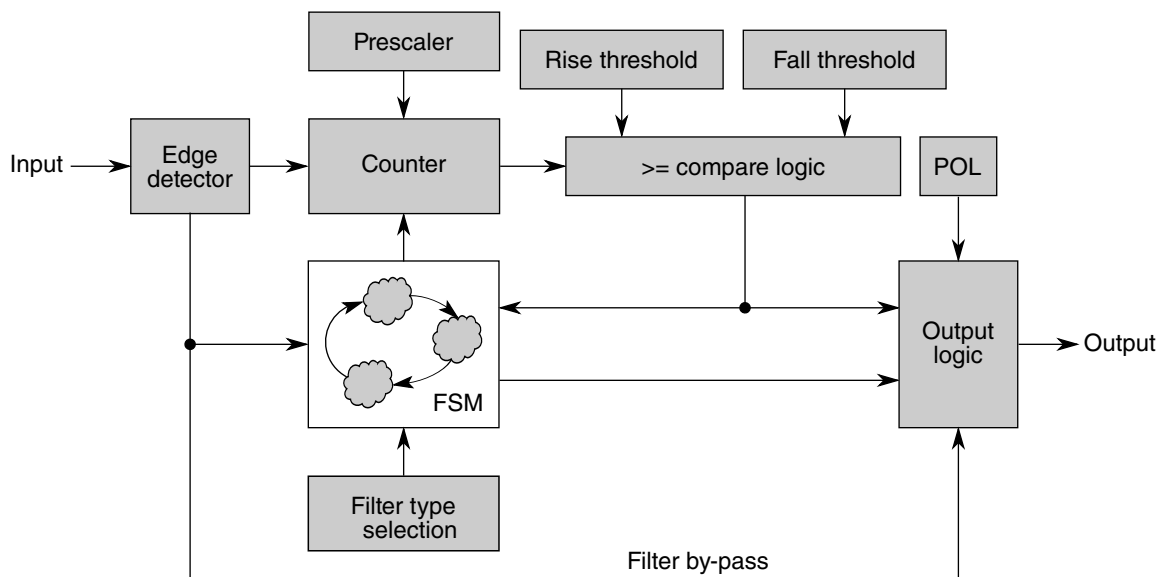


Figure 36-1. Input Glitch Filter block diagram

### 36.4 Input Glitch Filter overview

The Input Glitch Filter receives an input signal usually from a device PAD. Signals generated externally to the device are affected by environmental noise or glitches. The signal level has to be evaluated with some criteria in order to remove the noise which could affect the measurements done over this signal, such as transition time, period and level.

In a typical application, the Input Glitch Filter interfaces with one or more timer peripherals.

The strategy to remove glitches from an input signal is to monitor the level of the signal during a wide time window thus the glitches can be evaluated and removed.

The input signal should be stable for a certain amount of time before it can be used as a trigger or by a timer input channel. This strategy implies a delay on detecting a transition in the signal.

There is another strategy which is based on removing the glitches by introducing a blanking period after an edge is detected. In this strategy there is no delay from the edge detection in the input signal to the filtered signal, in the filter output. The strategy also allows input signals with different filter periods to be measured at the same time.

### 36.4.1 Interaction with other peripherals

The figure below shows an example of how the Input Glitch Filter interacts with on-chip peripherals.

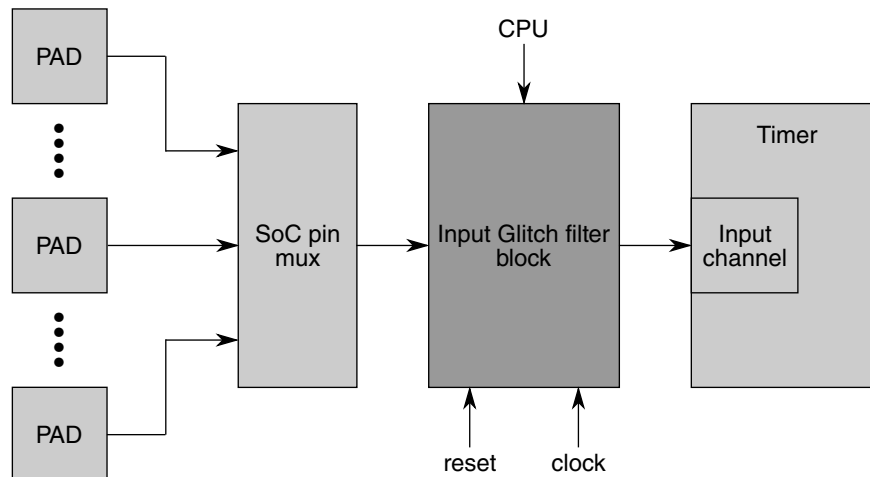


Figure 36-2. Input Glitch Filter application example

### 36.4.2 Mode of operation

The Input Glitch Filter has the following modes of operation:

- Configuration mode
- Normal mode
- Debug mode
- Low power mode

#### 36.4.2.1 Configuration mode

The configuration mode is the initial mode of the Input Glitch filter after system reset is applied. The configuration mode is selected when FGEN = '0' in the MCR register. The filter parameters such as rising and falling thresholds can be programmed when the filter

is in configuration mode without disturbing the internal device logic since the filter output is kept at logic '0' state. Once FGEN is programmed with logic '1' the filter enters the normal operating mode and is able to initiate the filtering process.

#### NOTE

The MDIS bit in the MCR register, if set to logic '1', makes the filter transition to low power mode independent of FGEN bit settings. Thus, in order to enter the configuration mode MDIS bit needs to be at logic '0'.

### 36.4.2.2 Normal mode

The normal mode of operation is entered when MCR register FGEN = '1'. In this mode the filter implements one of the filter types:

- Bypass filter
- Windowing filter
- Integrating filter
- Integrating-hold filter

These filter types are described in the following sections. Before entering the normal mode, the filter parameters, such as rise and fall thresholds, need to be programmed in the appropriate registers, as well as the type needs to be defined in the corresponding register.

#### NOTE

In the normal mode it is still possible to change the filter parameters, such as rise and fall thresholds. See section [On-the-fly filter reconfiguration](#) for more details.

### 36.4.2.3 Debug mode

The debug mode is entered when the MCR FRZ bit is set and a system level debug request occurs. Once the debug mode is entered, the internal filter counter is halted and the debug functions controlled by the FBP, FOH, and FOL bits are enabled. The filter output is frozen and remains at the state it was immediately before the debug mode was entered. The FOH and FOL MCR bits can be used to force the filter output to logic '1' or '0', respectively. These bits are self-negating, thus they always read as zero.



By setting the MCR FBP bit, the filter input state is copied to the filter output, thus allowing to control the filter output based on the filter input state. The output is forced during one clock cycle. After that the filter output remains at the new forced state until it is changed again by using a debug control bit or if the filter returns to the normal mode of operation.

### NOTE

The system level debug request occurs when a specific event, or a series of events, occurs in one of the device sub-modules and is interpreted by the device debug logic as a debug request. The Input Glitch Filter is not capable of generating an event that causes the assertion of the system level debug request.

#### 36.4.2.4 Low power mode

The low power mode is controlled by the MDIS bit in the MCR register or by the STOP signal which is the system level low power control signal. When entering low power mode the filter output remains unchanged at the state it was just before the low power mode was entered. Since the Input Glitch Filter clocks are stopped in this mode, the internal counter remains inactive. The description of this mode is covered in the Low Power Mode section of this document.

## 36.5 Memory map

The Input Glitch Filter registers are implemented in 32-bit. 8-bit write operations are allowed. The coherency in case of 8-bit access must be provided by the application software since there is no coherence mechanism built into the hardware.

### NOTE

RESERVED bit registers, when available for write operations, should be written always with logic '0'. Read access does not return meaningful data from RESERVED bit registers.

#### IGF memory map

| Address offset (hex) | Register name                            | Width (in bits) | Access | Reset value | Section/ page               |
|----------------------|------------------------------------------|-----------------|--------|-------------|-----------------------------|
| 0                    | Module Configuration Register (IGF_MCR0) | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| 4                    | Module Status Register (IGF_MSR0)        | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |

*Table continues on the next page...*

## IGF memory map (continued)

| Address offset (hex) | Register name                               | Width (in bits) | Access | Reset value | Section/ page               |
|----------------------|---------------------------------------------|-----------------|--------|-------------|-----------------------------|
| 10                   | Prescaler Register (IGF_PRESR0)             | 32              | R/W    | 0000_0000h  | <a href="#">36.5.3/1654</a> |
| 1C                   | Rising edge Threshold Register (IGF_RTHR0)  | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| 20                   | Falling edge Threshold Register (IGF_FTHR)  | 32              | R/W    | 0000_0000h  | <a href="#">36.5.5/1655</a> |
| 40                   | Module Configuration Register (IGF_MCR1)    | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| 44                   | Module Status Register (IGF_MSR1)           | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| 5C                   | Rising edge Threshold Register (IGF_RTHR1)  | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| 80                   | Module Configuration Register (IGF_MCR2)    | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| 84                   | Module Status Register (IGF_MSR2)           | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| 9C                   | Rising edge Threshold Register (IGF_RTHR2)  | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| C0                   | Module Configuration Register (IGF_MCR3)    | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| C4                   | Module Status Register (IGF_MSR3)           | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| DC                   | Rising edge Threshold Register (IGF_RTHR3)  | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| 100                  | Module Configuration Register (IGF_MCR4)    | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| 104                  | Module Status Register (IGF_MSR4)           | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| 11C                  | Rising edge Threshold Register (IGF_RTHR4)  | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| 140                  | Module Configuration Register (IGF_MCR5)    | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| 144                  | Module Status Register (IGF_MSR5)           | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| 15C                  | Rising edge Threshold Register (IGF_RTHR5)  | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| 180                  | Module Configuration Register (IGF_MCR6)    | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| 184                  | Module Status Register (IGF_MSR6)           | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| 19C                  | Rising edge Threshold Register (IGF_RTHR6)  | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| 1C0                  | Module Configuration Register (IGF_MCR7)    | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| 1C4                  | Module Status Register (IGF_MSR7)           | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| 1DC                  | Rising edge Threshold Register (IGF_RTHR7)  | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| 200                  | Module Configuration Register (IGF_MCR8)    | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| 204                  | Module Status Register (IGF_MSR8)           | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| 21C                  | Rising edge Threshold Register (IGF_RTHR8)  | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| 240                  | Module Configuration Register (IGF_MCR9)    | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| 244                  | Module Status Register (IGF_MSR9)           | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| 25C                  | Rising edge Threshold Register (IGF_RTHR9)  | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| 280                  | Module Configuration Register (IGF_MCR10)   | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| 284                  | Module Status Register (IGF_MSR10)          | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| 29C                  | Rising edge Threshold Register (IGF_RTHR10) | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| 2C0                  | Module Configuration Register (IGF_MCR11)   | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| 2C4                  | Module Status Register (IGF_MSR11)          | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| 2DC                  | Rising edge Threshold Register (IGF_RTHR11) | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| 300                  | Module Configuration Register (IGF_MCR12)   | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| 304                  | Module Status Register (IGF_MSR12)          | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| 31C                  | Rising edge Threshold Register (IGF_RTHR12) | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |

Table continues on the next page...

## IGF memory map (continued)

| Address offset (hex) | Register name                               | Width (in bits) | Access | Reset value | Section/ page               |
|----------------------|---------------------------------------------|-----------------|--------|-------------|-----------------------------|
| 340                  | Module Configuration Register (IGF_MCR13)   | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| 344                  | Module Status Register (IGF_MSR13)          | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| 35C                  | Rising edge Threshold Register (IGF_RTHR13) | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| 380                  | Module Configuration Register (IGF_MCR14)   | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| 384                  | Module Status Register (IGF_MSR14)          | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| 39C                  | Rising edge Threshold Register (IGF_RTHR14) | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| 3C0                  | Module Configuration Register (IGF_MCR15)   | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| 3C4                  | Module Status Register (IGF_MSR15)          | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| 3DC                  | Rising edge Threshold Register (IGF_RTHR15) | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| 400                  | Module Configuration Register (IGF_MCR16)   | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| 404                  | Module Status Register (IGF_MSR16)          | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| 41C                  | Rising edge Threshold Register (IGF_RTHR16) | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| 440                  | Module Configuration Register (IGF_MCR17)   | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| 444                  | Module Status Register (IGF_MSR17)          | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| 45C                  | Rising edge Threshold Register (IGF_RTHR17) | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| 480                  | Module Configuration Register (IGF_MCR18)   | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| 484                  | Module Status Register (IGF_MSR18)          | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| 49C                  | Rising edge Threshold Register (IGF_RTHR18) | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| 4C0                  | Module Configuration Register (IGF_MCR19)   | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| 4C4                  | Module Status Register (IGF_MSR19)          | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| 4DC                  | Rising edge Threshold Register (IGF_RTHR19) | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| 500                  | Module Configuration Register (IGF_MCR20)   | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| 504                  | Module Status Register (IGF_MSR20)          | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| 51C                  | Rising edge Threshold Register (IGF_RTHR20) | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| 540                  | Module Configuration Register (IGF_MCR21)   | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| 544                  | Module Status Register (IGF_MSR21)          | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| 55C                  | Rising edge Threshold Register (IGF_RTHR21) | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| 580                  | Module Configuration Register (IGF_MCR22)   | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| 584                  | Module Status Register (IGF_MSR22)          | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| 59C                  | Rising edge Threshold Register (IGF_RTHR22) | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| 5C0                  | Module Configuration Register (IGF_MCR23)   | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| 5C4                  | Module Status Register (IGF_MSR23)          | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| 5DC                  | Rising edge Threshold Register (IGF_RTHR23) | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| 600                  | Module Configuration Register (IGF_MCR24)   | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| 604                  | Module Status Register (IGF_MSR24)          | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| 61C                  | Rising edge Threshold Register (IGF_RTHR24) | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| 640                  | Module Configuration Register (IGF_MCR25)   | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| 644                  | Module Status Register (IGF_MSR25)          | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| 65C                  | Rising edge Threshold Register (IGF_RTHR25) | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |

Table continues on the next page...

## IGF memory map (continued)

| Address offset (hex) | Register name                               | Width (in bits) | Access | Reset value | Section/ page               |
|----------------------|---------------------------------------------|-----------------|--------|-------------|-----------------------------|
| 680                  | Module Configuration Register (IGF_MCR26)   | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| 684                  | Module Status Register (IGF_MSR26)          | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| 69C                  | Rising edge Threshold Register (IGF_RTHR26) | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| 6C0                  | Module Configuration Register (IGF_MCR27)   | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| 6C4                  | Module Status Register (IGF_MSR27)          | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| 6DC                  | Rising edge Threshold Register (IGF_RTHR27) | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| 700                  | Module Configuration Register (IGF_MCR28)   | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| 704                  | Module Status Register (IGF_MSR28)          | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| 71C                  | Rising edge Threshold Register (IGF_RTHR28) | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| 740                  | Module Configuration Register (IGF_MCR29)   | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| 744                  | Module Status Register (IGF_MSR29)          | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| 75C                  | Rising edge Threshold Register (IGF_RTHR29) | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| 780                  | Module Configuration Register (IGF_MCR30)   | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| 784                  | Module Status Register (IGF_MSR30)          | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| 79C                  | Rising edge Threshold Register (IGF_RTHR30) | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| 7C0                  | Module Configuration Register (IGF_MCR31)   | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| 7C4                  | Module Status Register (IGF_MSR31)          | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| 7DC                  | Rising edge Threshold Register (IGF_RTHR31) | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| 800                  | Module Configuration Register (IGF_MCR32)   | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| 804                  | Module Status Register (IGF_MSR32)          | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| 810                  | Prescaler Register (IGF_PRESR32)            | 32              | R/W    | 0000_0000h  | <a href="#">36.5.3/1654</a> |
| 81C                  | Rising edge Threshold Register (IGF_RTHR32) | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| 840                  | Module Configuration Register (IGF_MCR33)   | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| 844                  | Module Status Register (IGF_MSR33)          | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| 85C                  | Rising edge Threshold Register (IGF_RTHR33) | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| 880                  | Module Configuration Register (IGF_MCR34)   | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| 884                  | Module Status Register (IGF_MSR34)          | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| 89C                  | Rising edge Threshold Register (IGF_RTHR34) | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| 8C0                  | Module Configuration Register (IGF_MCR35)   | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| 8C4                  | Module Status Register (IGF_MSR35)          | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| 8DC                  | Rising edge Threshold Register (IGF_RTHR35) | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| 900                  | Module Configuration Register (IGF_MCR36)   | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| 904                  | Module Status Register (IGF_MSR36)          | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| 91C                  | Rising edge Threshold Register (IGF_RTHR36) | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| 940                  | Module Configuration Register (IGF_MCR37)   | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| 944                  | Module Status Register (IGF_MSR37)          | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| 95C                  | Rising edge Threshold Register (IGF_RTHR37) | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| 980                  | Module Configuration Register (IGF_MCR38)   | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| 984                  | Module Status Register (IGF_MSR38)          | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |

Table continues on the next page...

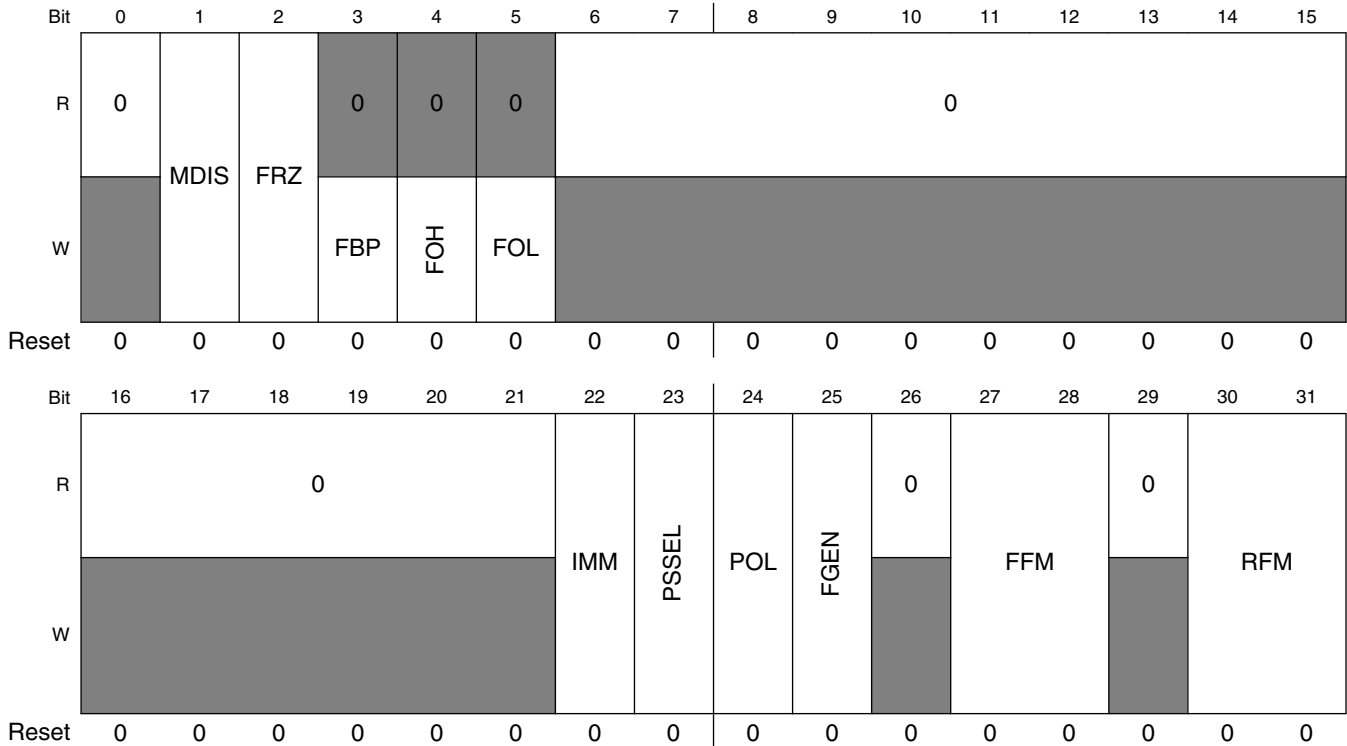
## IGF memory map (continued)

| Address offset (hex) | Register name                               | Width (in bits) | Access | Reset value | Section/ page               |
|----------------------|---------------------------------------------|-----------------|--------|-------------|-----------------------------|
| 99C                  | Rising edge Threshold Register (IGF_RTHR38) | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| 9C0                  | Module Configuration Register (IGF_MCR39)   | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| 9C4                  | Module Status Register (IGF_MSR39)          | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| 9DC                  | Rising edge Threshold Register (IGF_RTHR39) | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| A00                  | Module Configuration Register (IGF_MCR40)   | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| A04                  | Module Status Register (IGF_MSR40)          | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| A1C                  | Rising edge Threshold Register (IGF_RTHR40) | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| A40                  | Module Configuration Register (IGF_MCR41)   | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| A44                  | Module Status Register (IGF_MSR41)          | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| A5C                  | Rising edge Threshold Register (IGF_RTHR41) | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| A80                  | Module Configuration Register (IGF_MCR42)   | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| A84                  | Module Status Register (IGF_MSR42)          | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| A9C                  | Rising edge Threshold Register (IGF_RTHR42) | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| AC0                  | Module Configuration Register (IGF_MCR43)   | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| AC4                  | Module Status Register (IGF_MSR43)          | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| ADC                  | Rising edge Threshold Register (IGF_RTHR43) | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| B00                  | Module Configuration Register (IGF_MCR44)   | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| B04                  | Module Status Register (IGF_MSR44)          | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| B1C                  | Rising edge Threshold Register (IGF_RTHR44) | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |
| B40                  | Module Configuration Register (IGF_MCR45)   | 32              | R/W    | 0000_0000h  | <a href="#">36.5.1/1650</a> |
| B44                  | Module Status Register (IGF_MSR45)          | 32              | R/W    | 0000_0000h  | <a href="#">36.5.2/1653</a> |
| B5C                  | Rising edge Threshold Register (IGF_RTHR45) | 32              | R/W    | 0000_0000h  | <a href="#">36.5.4/1655</a> |

36.5.1 Module Configuration Register (IGF\_MCRn)

The Module Configuration register (MCR) provides access to low power, debug, and filter control bits.

Address: 0h base + 0h offset + (64d × i), where i=0d to 45d



IGF\_MCRn field descriptions

| Field         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 1<br>MDIS     | Module Disable<br><br>The MDIS bit forces the filter to enter a low power state. After this bit is asserted the clock for the module is removed thus filter operation is not possible. The access to filter registers is allowed.<br><br>0   Input Glitch Filter is not in low power mode due to MDIS bit<br>1   Input Glitch Filter is in low power mode                                                                                                                                                                                |
| 2<br>FRZ      | Freeze bit for debug operation<br><br>The FRZ bit enables the Input Glitch Filter to enter freeze mode (debug mode) when a system level signal request the module enters freeze (debug) mode. Once in freeze mode the filter operation is disabled meaning that transitions on the input signal will not propagate through the filter output. The bypass FBP bit can be used by the software to force the filter output to be the same as the filter input. In this mode FOH and FOL also can be used to define the filter output state. |

Table continues on the next page...

## IGF\_MCRn field descriptions (continued)

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                  | 0 Freeze mode disabled<br>1 Freeze mode enabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 3<br>FBP         | Force Bypass<br><br>This bit is self-negated thus is only read as zero. Writing 1 to this bit while in freeze mode forces the filter to bypass the value in the input to the filter output, thus allowing a software control on the filter. This bit is functional only in freeze mode.<br><br>0 no filter bypass<br>1 transfer input signal value to the filter output                                                                                                                                                                                                                                   |
| 4<br>FOH         | Force filter output high<br><br>If set, this bit forces the Input Glitch Filter output to '1'. This bit is functional only in debug mode. It is self-negated thus always read as zero.<br><br>0 no action<br>1 forces the filter output to '1'                                                                                                                                                                                                                                                                                                                                                            |
| 5<br>FOL         | Force filter output low<br><br>If set this bit forces the Input Glitch Filter output to '0'. This bit is functional only in debug mode. It is self-negated thus always read as zero.<br><br>0 no action<br>1 forces the filter output to '0'                                                                                                                                                                                                                                                                                                                                                              |
| 6–21<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 22<br>IMM        | Immediate edge propagation control bit<br><br>The IMM bit controls the propagation of an edge through the filter. If asserted, this bit defines that an edge at the filter input propagates through the filter output independent of the prescaler settings. In bypass mode the propagation delay is three system clock cycles. If negated, the signal propagation depends upon the prescaler settings. The filter hardware will only allow changes to this bit if the filter is disabled, FGEN = 0<br><br>0 edge propagation depends on prescaler<br>1 edge propagation within three system clock cycles |
| 23<br>PSSEL      | Prescaler selection bit<br><br>The PSSEL selects if the prescaler used by the Input Glitch Filter is from internal source (internal prescaler counter) or from external source. The filter hardware will only allow changes to this bit if the filter is disabled, FGEN=0.<br><br>0 internal prescaler selected<br>1 external prescaler selected                                                                                                                                                                                                                                                          |
| 24<br>POL        | Output polarity bit<br><br>This bit sets the polarity of the filter output to inverted or not inverted. The polarity control implemented by the POL bit is applied at the filter output stage thus also changing the reset state of the output, even if the filter is disabled by the FGEN bit. The filter hardware will only allow changes to this bit if the filter is disabled, FGEN=0.                                                                                                                                                                                                                |

*Table continues on the next page...*

IGF\_MCR<sub>n</sub> field descriptions (continued)

| Field          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | 0 not inverted output<br>1 inverted output                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 25<br>FGEN     | Filter global enable<br><br>This control bit enables the filter operation, meaning that if the filter is enabled the filtering selected types are applied to the rising and falling edges of the input signal. If the filter is not enabled, then the filter output remains unchanged independent of the filter input signal. Note that the filter output state is affected by the POL bit when the filter is disabled.<br><br>0 filter is disabled: filter output keeps current state.<br>1 filter is enabled: filtering is applied to selected edges.   |
| 26<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 27–28<br>FFM   | Falling edge filter type selection<br><br>This bit field selects the filter type for the filter input signal falling edge. The filter hardware will only allow changes to this bit if the filter is disabled, FGEN=0.<br><br>00 bypass: edge is propagated to filter output without any filtering in three system clock cycles<br>01 windowing: windowing filter is selected for the falling edge<br>10 integrating: integrating filter is selected for the falling edge<br>11 integrating-hold: integrating-hold filter is selected for the falling edge |
| 29<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 30–31<br>RFM   | Rising edge filter type selection<br><br>This bit field selects the filter type for the input signal rising edge. The filter hardware will only allow changes to this bit if the filter is disabled, FGEN=0.<br><br>00 bypass: edge is propagated to filter output without any filtering in three system clock cycles<br>01 windowing: windowing filter is selected for the rising edge<br>10 integrating: integrating filter is selected for the rising edge<br>11 integrating-hold: integrating-hold filter is selected for the rising edge             |



### 36.5.2 Module Status Register (IGF\_MSRn)

The Module status register (MSR) provides access to filter status bits and corresponding clear bits as well as access to the state of filter input and output signals.

Address: 0h base + 4h offset + (64d × i), where i=0d to 45d

|       |    |    |    |    |    |    |    |    |    |    |       |       |       |       |     |     |
|-------|----|----|----|----|----|----|----|----|----|----|-------|-------|-------|-------|-----|-----|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10    | 11    | 12    | 13    | 14  | 15  |
| R     | 0  |    |    |    |    |    |    |    |    |    |       |       |       |       |     |     |
| W     |    |    |    |    |    |    |    |    |    |    |       |       |       |       |     |     |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0     | 0     | 0     | 0     | 0   | 0   |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26    | 27    | 28    | 29    | 30  | 31  |
| R     | 0  |    |    |    |    |    |    |    |    |    | WEDGE | FEDGE | RNDET | ENDET | FLI | FLO |
| W     |    |    |    |    |    |    |    |    |    |    | w1c   | w1c   | w1c   | w1c   |     |     |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0     | 0     | 0     | 0     | 0   | 0   |

**IGF\_MSRn field descriptions**

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–25<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 26<br>WEDGE      | Filter is active waiting for an edge<br><br>This bit is set when the filter is enabled but the internal counter is not active. It indicates that the filter is waiting for an edge to occur at the filter input. It is set by the Input Glitch Filter hardware and cleared by the software by writing 1 to the bit. Since the intention of this bit is to capture the dynamic of the filter hardware, it should be cleared after the read thus being able to capture a new event.<br><br>0 Input Glitch Filter is disabled or the internal counter is active<br>1 Input Glitch Filter is waiting for an edge |
| 27<br>FEDGE      | Filter is active processing an edge<br><br>This bit is set during the filter active state thus when the filter internal counter is counting, processing an edge in a filtering mode. It is cleared by the software by writing 1 to the bit. Since the intention of this bit is to capture the dynamic of the filter hardware, it should be cleared after the read thus being able to capture a new event.<br><br>0 Input Glitch Filter internal counter is not active<br>1 Input Glitch Filter internal counter is active                                                                                    |

*Table continues on the next page...*

IGF\_MSR<sub>n</sub> field descriptions (continued)

| Field       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 28<br>RNDET | <p>Rise noise detected bit</p> <p>This bit indicates that the filter filtered out noise in the input signal after a rising edge was detected. The criteria to set this bit is when a rising edge is detected thus starting the filter counter and before this edge pass through the filter (if filter is in integrating filter types) or during a window (if filter is in windowing filter types) another edge occurs in the input signal. This bit is set by a noise detection and cleared by the software by writing 1 to this bit.</p> <p>0 noise was not detected<br/>1 noise was detected after a rising edge</p> |
| 29<br>FNDET | <p>Fall noise detected bit</p> <p>This bit indicates that the filter filtered out noise in the input signal after a falling edge was detected. The criteria to set this bit is when a falling edge is detected, thus starting the filter counter, and, if the filter is in integrating types, before this edge passes through the filter, or, in windowing types, when another edge occurs in the input signal during a window. This bit is set by a noise detection and cleared by the software by writing 1 to this bit.</p> <p>0 noise was not detected<br/>1 noise was detected after a falling edge</p>           |
| 30<br>FLI   | <p>Filter input</p> <p>The FLI bit follows the state of the filter input value as captured by the filter input logic, which is affected by the IMM bit and prescaler settings.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 31<br>FLO   | <p>Filter output</p> <p>The FLO bit follows the state of the filter output value considering the POL bit state.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

36.5.3 Prescaler Register (IGF\_PRESR<sub>n</sub>)

The prescaler register (PRESR) provides access to the prescaler bits.

Address: 0h base + 10h offset + (2048d × i), where i=0d to 1d

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|------|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22   | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    | FPRE |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

IGF\_PRESR<sub>n</sub> field descriptions

| Field            | Description                                                                                                                                                                                            |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–22<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                     |
| 23–31<br>FPRE    | <p>Filter prescaler</p> <p>The prescaler defines the rate of the filter counter FCOUNT. The filter counter clock rate is defined by the formula: Filter counter clock = module clock / (FPRE + 1).</p> |

### 36.5.4 Rising edge Threshold Register (IGF\_RTHRn)

The Rising edge Threshold register (RTHR) provides access to the rising edge threshold bits.

Address: 0h base + 1Ch offset + (64d × i), where i=0d to 45d

|       |   |   |   |   |   |   |   |   |     |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|-----|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8   | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   | RTH |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |     |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**IGF\_RTHRn field descriptions**

| Field           | Description                                                                                                               |
|-----------------|---------------------------------------------------------------------------------------------------------------------------|
| 0–7<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                   |
| 8–31<br>RTH     | Rising edge threshold<br>This bit field defines the filter counter threshold when a rising edge occurs is being filtered. |

### 36.5.5 Falling edge Threshold Register (IGF\_FTTHR)

The Falling edge Threshold register (FTTHR) provides access to the falling edge threshold bits.

Address: 0h base + 20h offset = 20h

|       |   |   |   |   |   |   |   |   |     |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|-----|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8   | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   | FTH |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |     |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**IGF\_FTTHR field descriptions**

| Field           | Description                                                                                                          |
|-----------------|----------------------------------------------------------------------------------------------------------------------|
| 0–7<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                              |
| 8–31<br>FTH     | Falling edge threshold<br>This bit field defines the filter counter threshold when a falling edge is being filtered. |

## 36.6 Functional description

The following sections describe the Input Glitch Filter module functionality.

### 36.6.1 Input double sampling

The filter input is double sampled for clock domain translation thus avoiding asynchronous signals to propagate through the filter internal logic. This stage is composed by two flip-flops connected back to back. The delay caused by this logic is therefore two clock cycles.

Note that due to the synchronous nature of the Input Glitch Filter design, if a glitch smaller than one clock cycle occurs at the filter input it will not propagate to the filter internal logic and thus will not propagate to the filter output. Also, if clock is not applied to the filter (i.e. in low power mode) any transition at the filter input will not propagate to the filter output.

### 36.6.2 Filter types

The Input Glitch Filter implements three main filter types:

- Windowing
- Integrating
- Integrating hold

These filter types can be selected for the rising and falling edges independently, thus allowing different filter types to be selected for different edges (i.e. rising edge filtered with windowing filter and falling edge filtered with integrating).

#### 36.6.2.1 Windowing filter

This filter type detects an edge (positive or negative edge) at the filter input and then mask out detection of subsequent edges over a period of time, which is defined as the blanking period, thus effectively behaving as a high pass filter. After an edge is detected the level at the filter input is propagated to the filter output within three clock cycles. The delay is caused by the synchronization stage at the input and by the filter output register stage. During the blanking period any transition at the filter input is disregarded. After the blanking period has elapsed, the filter compares the input level state with the filter output state. In case they are different a transition is inferred and re-evaluate according to the filter type assigned to that transition (positive or negative edges). Since, in this example, the windowing filter type is programmed for both edges, the inferred negative edge is propagated to the filter output and another blanking period is initiated. The figure below describes the windowing filter type behavior.

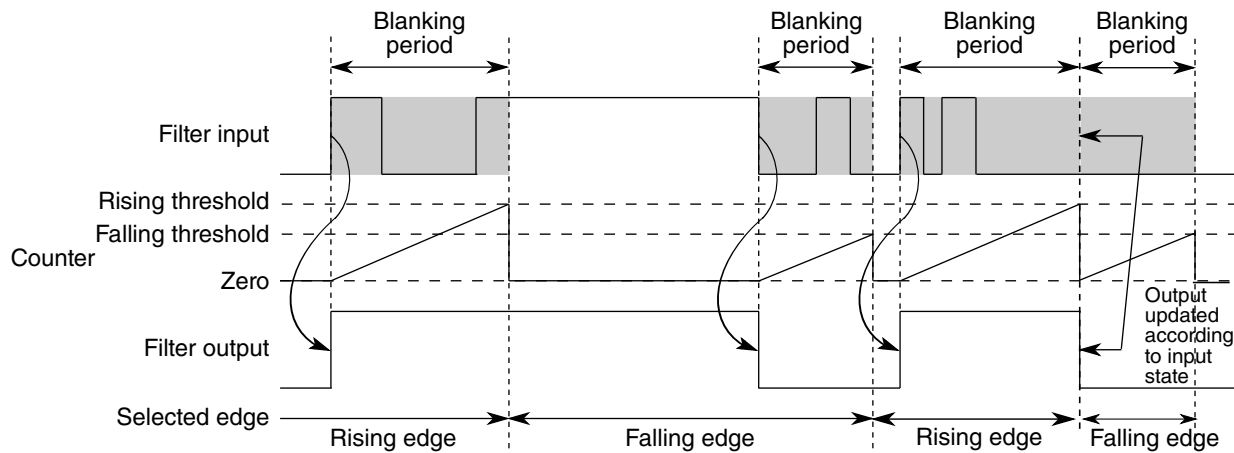


Figure 36-148. Windowing filter type

### 36.6.2.2 Integrating filter

The integrating filter behaves like a low-pass filter, thus removing high frequency bursts from the input signal. The input and output signals are compared and if they are at different states an internal filter counter is enabled.

The direction of the counter (up or down counting) is defined by the states of the input and output filter signals as follows:

- Filter output is at the same state as the filter input: down-counting
- Filter output is at different state as the filter input: up-counting

The counter is compared with the rising or falling threshold register according to:

- Rising threshold register is selected if the rising edge is being filtered.
- Falling threshold register is selected if the falling edge is being filtered.

The filter output changes state to the same state as the filter input:

- If filter counter reaches the selected threshold for the edge being filtered.

The internal counter is disabled and cleared:

- If filter counter reaches the selected threshold for the edge being filtered.

The figure below describes an example of the Integrating filter type operation.

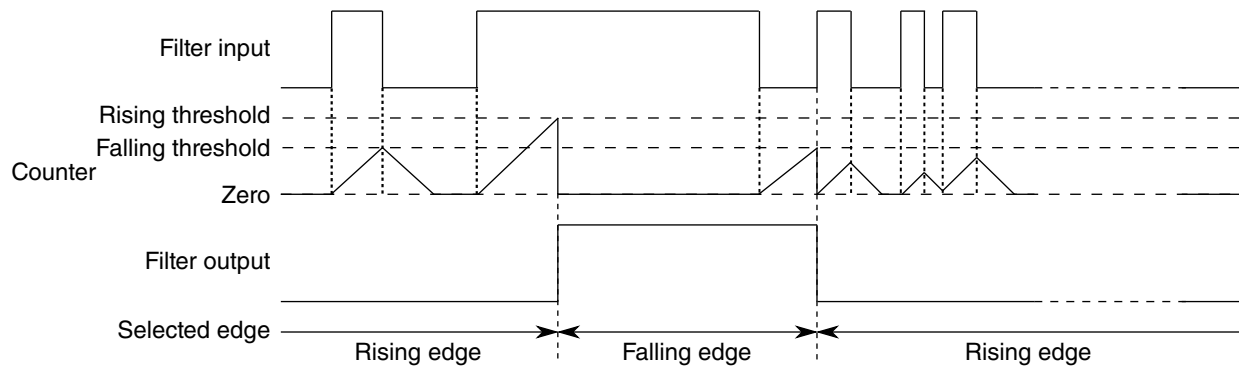


**Figure 36-149. Integrating filter type**

### 36.6.2.3 Integrating-hold filter

The integrating-hold filter type is similar to the integrating filter. The difference is the counter, which only counts in up direction instead of up-down in the integrating mode. In this filter configuration the input and output signals are compared and if they are at different states the filter counter is enabled. This counter is controlled by the state of the filter input and filter output signals as follows:

- Filter output is at the same state as the filter input: counter is in hold state (no-counting).
- Filter output is at different state as the filter input: counter is counting up.

The counter is compared with the rising or falling threshold register according to:

- Rising threshold register is selected if the rising edge is being filtered.
- Falling threshold register is selected if the falling edge is being filtered.

The filter output changes state to the same state as the filter input:

- If the filter counter reaches the selected threshold for the edge being filtered.

The internal counter is disabled and cleared:

- If the filter counter reaches the selected threshold for the edge being filtered.

The figure below shows an example of the Integrating-hold filter type operation.

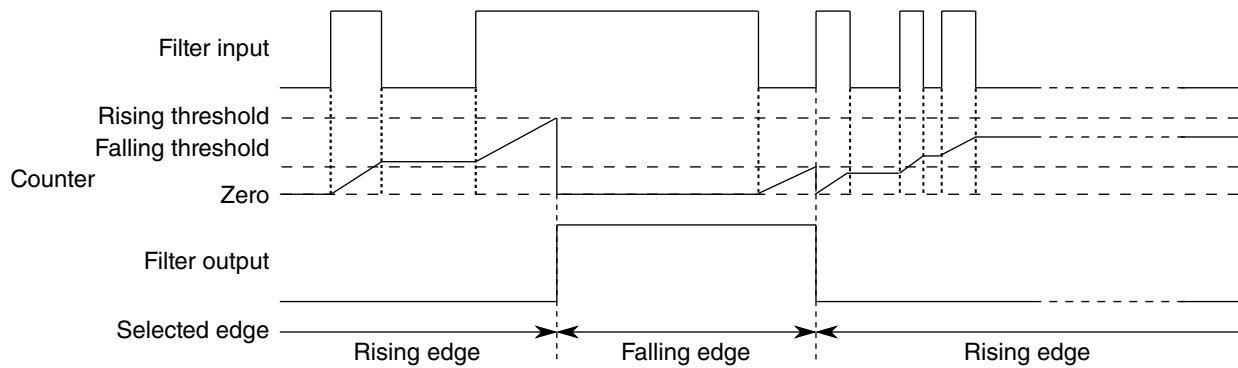


**Figure 36-150. Integrating-hold filter type**

### 36.6.2.4 Bypass filter

In the bypass filter type, the Input Glitch Filter immediately propagates any transition on the filter input to the filter output. This filter type does not provide any protection against glitches in the input signal. It is intended to be used in situations where the input signal is already filtered, or it is a glitch free signal, or the filter output is connected to a module that already has filtering capabilities. The bypass filter type is independently selected for rising or falling edges. Thus rising edges may be not filtered by the bypass filter type while the falling edges are filtered by any of the available filter types.

The bypass filter presents a delay in the edge propagation through the filter that is caused by the double sampling at the filter input and by the output filter register. Therefore, the total delay is three clock cycles.

### 36.6.2.5 Combined filter

The Input Glitch Filter allows the combination of filter types for different edges, positive and negative. As an example, the rising edges may be filtered using the windowing filter while the falling edges are filtered with an integrating filter. This configuration of the Input Glitch Filter is provided for situations where the noise present in the rising edges is different from the noise in the falling edges. Another use case is when the delay requirements of the input signal is different for rising and falling edges.

The figure below shows the combined filter operation. The windowing filter type is applied to the rising edges. The integrating filter is used for the falling edges.

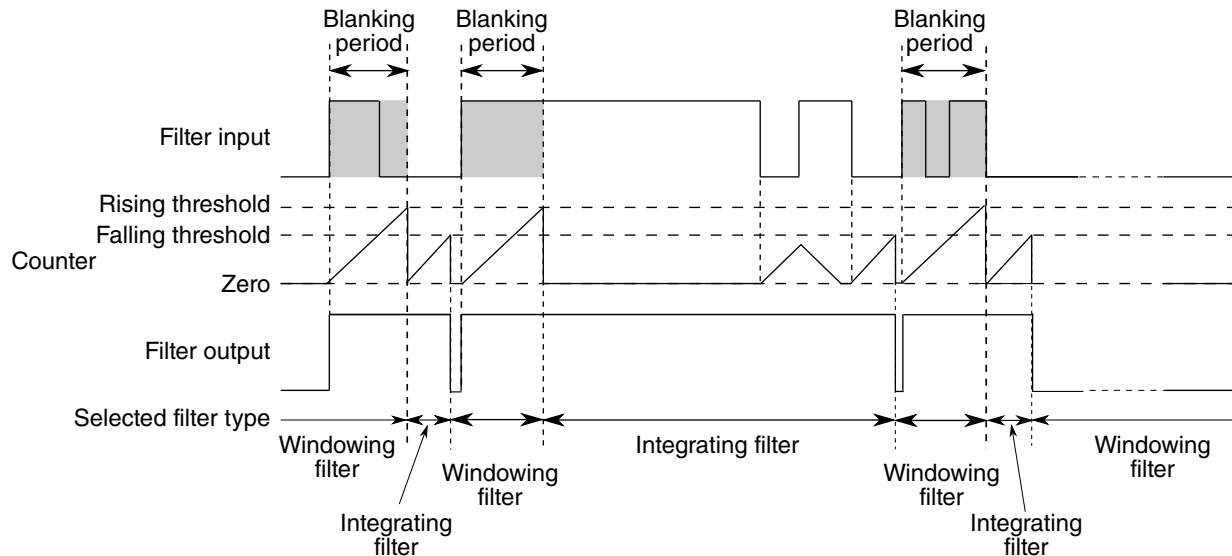


**Figure 36-151. Combined filter type (windowing + integrating)**

### 36.6.3 Application

This section describes several applications for the Input Glitch filter module.

#### 36.6.3.1 Application use case 1: integrating filter

The integrating filter type is particularly useful in applications where glitches occur spuriously while the signal is expected to be in a steady state (either 1 or 0). Choosing correct up and down filter clock rates can ensure that the effects of spurious glitches can be completely eliminated, whilst minimizing propagation delays.

The application must be able to tolerate delays between the actual edge occurrence and the time that the signal is propagated through the filter logic. Because the delays are constants defined by the detection thresholds, application software may compensate for them, provided little or no noise occurs on the detected edge.

#### 36.6.3.2 Application use case 2: windowing filter

The windowing filter type is particularly useful for removing noise bounce produced by a switch or sensor when its signal changes state. In this case, there is no propagation delay (except for the filter input synchronization logic and output register) between the filter input and filter output signals. This type of filter would not necessarily provide much immunity to spurious noise that occurred during the steady state portion of the signal. In



order to optimize the characteristics of this filter, the blanking period should be of the same magnitude as the expected input signal rate. Signals that change frequency can be accommodated by software by dynamically updating the blanking period.

### **36.6.3.3 Application use case 3: windowing filter**

Safety applications may need to compare the inputs obtained from different input pins, but derived from a single external source. This form of redundancy may detect non-common mode faults that may occur in the portions of the SoC that route and process the incoming signal. Having two different instances of windowing filter used on each input pin will ensure that the signal edge timing for both input paths is equal. Applying different filter characteristics for each input path can enhance the ability to detect SoC fault conditions.

### **36.6.3.4 Application use case 4: combined filter**

Some applications may have rising edge signals that exhibit different characteristics to falling edge signals which might require different filtering to optimize the filtering effect. For example the “first” rising edge may represent a true real time event with a fast slew rate that must be captured without delay, while the falling edge signal may have a slow slew rate with many noisy transitions. The Input Glitch Filter can apply, for example, a windowing filter on the rising edge to ensure no loss of real time information and an integrating filter on the falling edge to eliminate noise from being detected as a new rising transition.

### **36.6.3.5 Application use case 5: prescaler settings in windowing filter**

The prescaler settings influence the timing for an edge to propagate through the filter. In windowing filter the IMM bit in the MCR register controls the way an edge is propagated through the filter. The figure below describes the windowing filter operation with regard to the IMM bit settings.

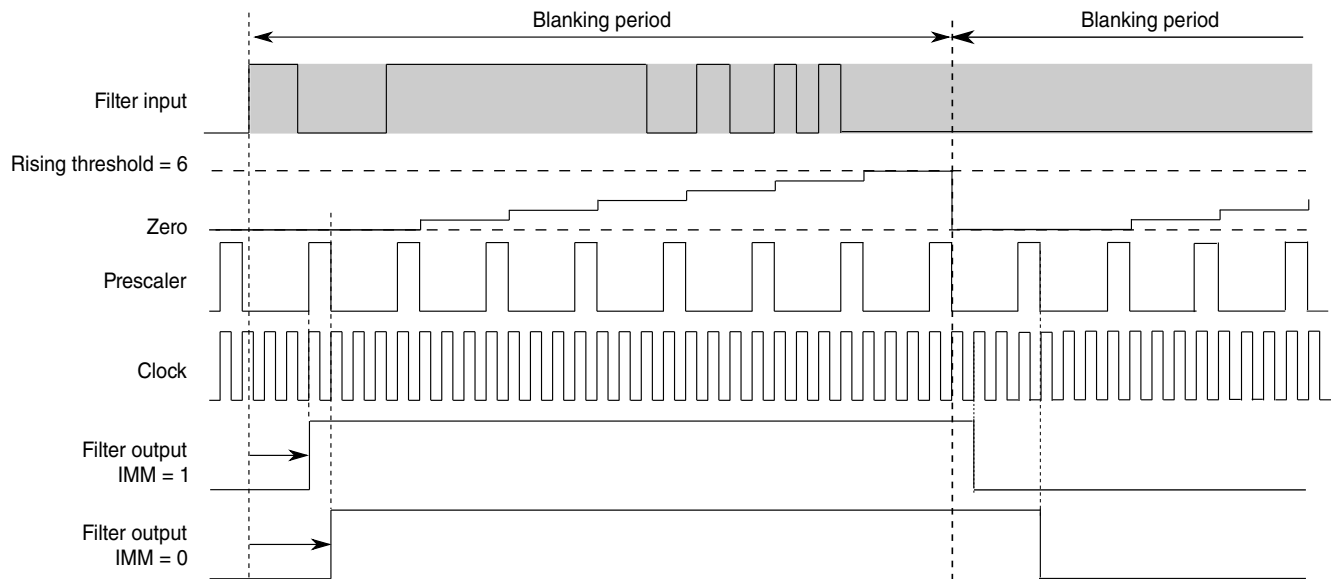


**Figure 36-152. Input signal propagation through the filter in windowing filter type**

If  $IMM = 0$  then the input signal edges are propagated according to the prescaler ticks. If  $IMM = 1$  then the input edges are propagated immediately (within 3 clock cycles) to the filter output.

### 36.6.3.6 Application use case 6: prescaler settings in bypass

The prescaler settings influence the timing for an input signal edge to propagate through the filter. In bypass, the  $IMM$  bit in the  $MCR$  register controls the way an edge is propagated through the filter. If  $IMM = 0$ , the prescaled clock is used to sample the input filter signal. The resolution the input signal is sampled is therefore related to the prescaler frequency. Glitches narrower than the prescaler period that are not aligned with the prescaler clock are not captured by the filter. If  $IMM = 1$ , the system clock is used to sample the filter input signal instead of the prescaled clock. In this case, the resolution of the sampled input signal is the system clock.

The figure below describes the filter input signal sampling behavior for  $IMM = 0$  and  $IMM = 1$ .

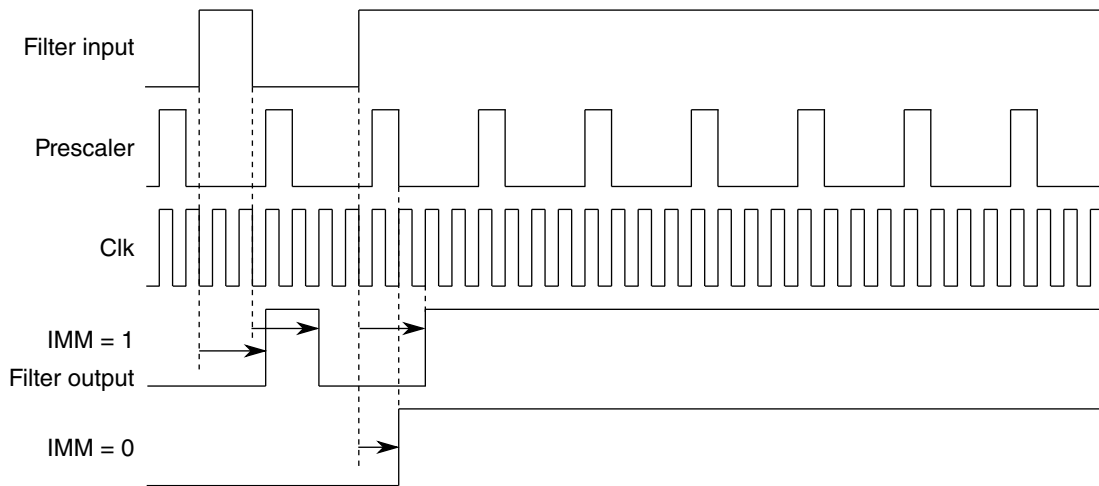


**Figure 36-153. Input signal propagation through the filter in bypass**

### 36.6.4 Programming guidelines

The following generalized procedure may be used to configure the Input Glitch Filter Block:

1. Program the SoC pin mux logic to route an external signal to the input of the selected glitch filter.
2. Enable the destination module to receive the Input Glitch filter output.
3. Enable the clock source at the SoC level.
4. Program the Glitch Filter counter prescaler(s) to the desired rate.
5. Select the filter type for each edge using MCR[FFM] and MCR[RFM].
6. Clear any status bits if necessary.
7. Enable the filter using MCR[FGEN]

### 36.6.5 On-the-fly filter reconfiguration

The architecture of the Input Glitch Filter allows for on the fly reconfiguration of filter parameters such as rise and fall threshold values.

If a parameter is changed it influences the filter operation immediately after the new parameter value was written. The main problem when changing parameters, such as a threshold value, when the filter is enabled is that the filter counter could lose a match due to the parameter change. A great or equal comparator is used in the threshold match logic which allows correct filter operation when the threshold parameters are changed. The figure below shows an example of changing the rising edge parameter value. Note that in this case the match occurs immediately after the parameter is changed since the new parameter value is below the current counter value.

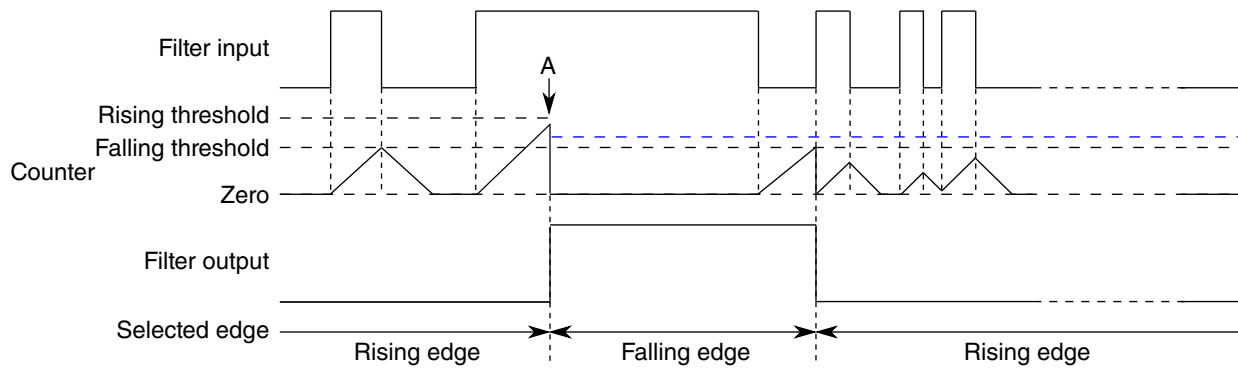


**Figure 36-154. On-the-fly rising threshold parameter change**

In the case shown in the figure above the match occurs immediately after the new rising threshold is programmed at 'A'.

### NOTE

Note that when changing the threshold values the match may not occur when the counter is equal to the threshold value (the old or the new programmed value). Instead, the match may occur in between the old and new programmed thresholds when the threshold re-programming instant A is done when the counter is greater than the new value. This is a transitory effect which may occur when the threshold value is changed and should be considered by the application.

## 36.7 Power saving mode

The Input Glitch Filter implements power saving mode thus allowing to reduce power consumption when device inputs connected to the filter are not used or the filter is not necessary to be functioning in a specific SoC mode of operation.

### 36.7.1 STOP mode

There are two ways for the Input Glitch filter to enter low power mode:

- MDIS bit
- STOP mode system request

The MDIS control bit in the MCR register, when set, disables the filter logic by stopping the module clock signals and thus halting the internal filter counter which reduces the dynamic power consumption of the module. The MDIS bit provides a selective control of the low power operation of each one of the Input Glitch Filter instances in the device.

The STOP mode system request is similar to MDIS operation with exception that it applies to all Input Glitch Filter instances in the device. It does not need to be enabled by any control bit in the Input Glitch filter module. The low power mode in this case is controlled by the assertion of the STOP signal which is a system wide low power control signal.



## **Chapter 37**

### **FlexCAN (CAN)**

#### **37.1 Chip-specific FlexCAN information**

This chip has four FlexCAN modules.

Each FlexCAN instance supports 64 message buffers.

##### **37.1.1 FlexCAN RAM**

On this chip, each FlexCAN module instance has four 1.5 KB dedicated RAM blocks.

##### **37.1.2 FlexCAN reset**

The FlexCAN architecture has two sources of reset: one resets the entire FlexCAN system and the other resets only the Host Control Interface (CHI). This chip supports only the global reset, which is driven by the internal reset of the device.

##### **37.1.3 CAN\_MCR[SUPV] and access to FlexCAN registers**

On this chip, regardless of the value of CAN\_MCR[SUPV], all core accesses to FlexCAN registers occur in supervisor mode.

#### **37.2 Introduction**

The FlexCAN module is a communication controller implementing the CAN protocol according to the CAN 2.0 B protocol specification. A general block diagram is shown in the following figure, which describes the main subblocks implemented in the FlexCAN module, including one associated memory for storing message buffers, Receive (Rx)

Global Mask registers, Receive Individual Mask registers, Receive FIFO filters, and Receive FIFO ID filters. The functions of the submodules are described in subsequent sections.

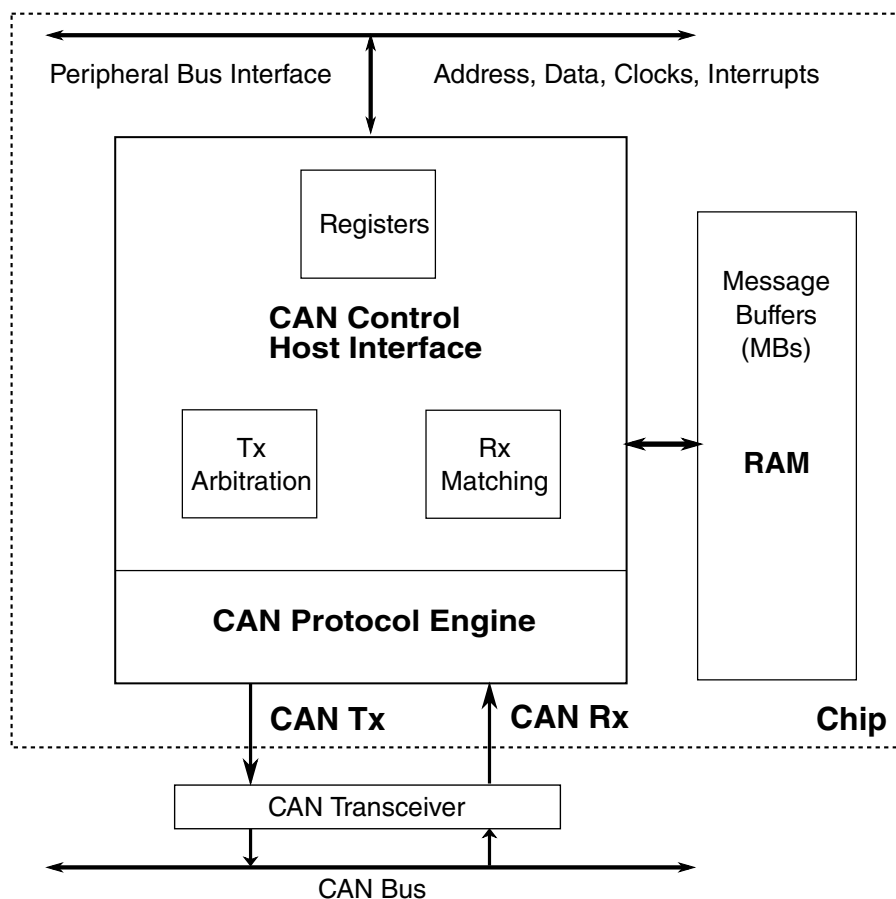


**Figure 37-1. FlexCAN block diagram**

### 37.2.1 Overview

The CAN protocol was primarily designed to be used as a vehicle serial data bus, meeting the specific requirements of this field:

- Real-time processing
- Reliable operation in the EMI environment of a vehicle
- Cost-effectiveness
- Required bandwidth

The FlexCAN module is a full implementation of the CAN protocol specification, Version 2.0 B, which supports both standard and extended message frames. The message buffers are stored in an embedded RAM dedicated to the FlexCAN module. See the chip-specific FlexCAN information for the actual number of message buffers configured in the chip.



The CAN Protocol Engine (PE) submodule manages the serial communication on the CAN bus:

- Requesting RAM access for receiving and transmitting message frames
- Validating received messages
- Performing error handling

The Controller Host Interface (CHI) sub-module handles message buffer selection for reception and transmission, taking care of arbitration and ID matching algorithms.

The Bus Interface Unit (BIU) sub-module controls the access to and from the internal interface bus, in order to establish connection to the CPU and to other blocks. Clocks, address and data buses, interrupt outputs and test signals are accessed through the BIU.

## 37.2.2 FlexCAN module features

The FlexCAN module includes these distinctive legacy features:

- Full implementation of the CAN protocol specification, Version 2.0 B
  - Standard data and remote frames
  - Extended data and remote frames
  - Zero to eight bytes data length
  - Programmable bit rate up to 1 Mb/sec
  - Content-related addressing
- Compliant with the ISO 11898-1 standard
- Flexible mailboxes of zero to eight bytes data length
- Each mailbox configurable as receive or transmit, all supporting standard and extended messages
- Individual Rx Mask registers per mailbox
- Full-featured Rx FIFO with storage capacity for up to six frames and automatic internal pointer handling
- Transmission abort capability
- Programmable clock source to the CAN Protocol Interface, either bus clock or crystal oscillator
- Unused structures space can be used as general purpose RAM space

- Listen-Only mode capability
- Programmable Loop-Back mode supporting self-test operation
- Programmable transmission priority scheme: lowest ID, lowest buffer number, or highest priority
- Time stamp based on 16-bit free-running timer
- Global network time, synchronized by a specific message
- Maskable interrupts
- Independence from the transmission medium (an external transceiver is assumed)
- Short latency time due to an arbitration scheme for high-priority messages
- Low power modes

New major features are also provided:

- Remote request frames may be handled automatically or by software
- CAN bit time settings and configuration bits can only be written in Freeze mode
- Tx mailbox status (Lowest priority buffer or empty buffer)
- Identifier Acceptance Filter Hit Indicator (IDHIT) register for received frames
- SYNCH bit available in Error in Status 1 register to inform that the module is synchronous with CAN bus
- CRC status for transmitted message
- Rx FIFO Global Mask register
- Selectable priority between mailboxes and Rx FIFO during matching process
- Powerful Rx FIFO ID filtering, capable of matching incoming IDs against either 128 extended, 256 standard, or 512 partial (8 bit) IDs, with up to 32 individual masking capability
- 100% backward compatibility with previous FlexCAN version
- Supports detection and correction of errors in memory read accesses. Each byte of FlexCAN memory is associated to 5 parity bits and the error correction mechanism assures that in this 13-bit word, errors in one bit can be corrected (corrected errors) and errors in 2 bits can be detected but not corrected (non-corrected errors).

### 37.2.3 Modes of operation

The FlexCAN module has these functional modes:

- Normal mode (User or Supervisor):

In Normal mode, the module operates receiving and/or transmitting message frames, errors are handled normally, and all CAN Protocol functions are enabled. User and Supervisor Modes differ in the access to some restricted control registers.

- Freeze mode:

Freeze mode is enabled when the FRZ bit in MCR is asserted. If enabled, Freeze mode is entered when MCR[HALT] is set or when Debug mode is requested at chip level and MCR[FRZ\_ACK] is asserted by the FlexCAN. In this mode, no transmission or reception of frames is done and synchronicity to the CAN bus is lost. See [Freeze mode](#) for more information.

- Listen-Only mode:

The module enters this mode when the LOM field in the Control 1 Register is asserted. In this mode, transmission is disabled, all error counters are frozen, and the module operates in a CAN Error Passive mode. Only messages acknowledged by another CAN station will be received. If FlexCAN detects a message that has not been acknowledged, it will flag a BIT0 error (without changing the REC), as if it was trying to acknowledge the message.

- Loop-Back mode:

The module enters this mode when the LPB field in the Control 1 Register is asserted. In this mode, FlexCAN performs an internal loop back that can be used for self-test operation. The bit stream output of the transmitter is internally fed back to the receiver input. The Rx CAN input pin is ignored and the Tx CAN output goes to the recessive state (logic '1'). FlexCAN behaves as it normally does when transmitting and treats its own transmitted message as a message received from a remote node. In this mode, FlexCAN ignores the bit sent during the ACK slot in the CAN frame acknowledge field to ensure proper reception of its own message. Both transmit and receive interrupts are generated.

For low-power operation, the FlexCAN module has:

- Module Disable mode:

This low-power mode is entered when the MDIS bit in the MCR Register is asserted by the CPU and the LPM\_ACK is asserted by the FlexCAN. When disabled, the module requests to disable the clocks to the CAN Protocol Engine and Controller Host Interface submodules. Exit from this mode is done by negating the MDIS bit in the MCR register. See [Module Disable mode](#) for more information.

- Doze mode:

This low power mode is entered when the DOZE bit in MCR is asserted and Doze mode is requested at chip level and the LPM\_ACK bit in the MCR Register is asserted by the FlexCAN. When in Doze mode, the module requests to disable the clocks to the CAN Protocol Engine and the CAN Controller-Host Interface submodules. Exit from this mode happens when the DOZE bit in MCR is negated, when the chip is removed from Doze mode. See [Doze mode](#) for more information.

- Stop mode:

This low power mode is entered when Stop mode is requested at chip level and the LPM\_ACK bit in the MCR Register is asserted by the FlexCAN. When in Stop Mode, the module puts itself in an inactive state and then informs the CPU that the clocks can be shut down globally. Exit from this mode happens when the Stop mode request is removed. See [Stop mode](#) for more information.

## 37.3 FlexCAN signal descriptions

The FlexCAN module has two I/O signals connected to the external chip pins. These signals are summarized in the following table and described in more detail in the next subsections.

**Table 37-1. FlexCAN signal descriptions**

| Signal | Description      | I/O    |
|--------|------------------|--------|
| CAN Rx | CAN Receive Pin  | Input  |
| CAN Tx | CAN Transmit Pin | Output |

### 37.3.1 CAN Rx

This pin is the receive pin from the CAN bus transceiver. Dominant state is represented by logic level 0. Recessive state is represented by logic level 1.

### 37.3.2 CAN Tx

This pin is the transmit pin to the CAN bus transceiver. Dominant state is represented by logic level 0. Recessive state is represented by logic level 1.

## 37.4 Memory map/register definition

This section describes the registers and data structures in the FlexCAN module. The base address of the module depends on the particular memory map of the chip.

### 37.4.1 FlexCAN memory mapping

The complete memory map for a FlexCAN module is shown in the following table.

The address space occupied by FlexCAN has 128 bytes for registers starting at the module base address, followed by embedded RAM starting at address 0x0080.

Each individual register is identified by its complete name and the corresponding mnemonic. The access type can be Supervisor (S) or Unrestricted (U). Most of the registers can be configured to have either Supervisor or Unrestricted access by programming the SUPV field in the MCR register. These registers are identified as S/U in the Access column of [Table 37-2](#).

#### NOTE

An invalid register access will result in a bus error. This includes reading a write-only register, writing a read-only register or accessing an invalid address.

**Table 37-2. Register access and reset information**

| Register                                     | Access type | Affected by hard reset | Affected by soft reset |
|----------------------------------------------|-------------|------------------------|------------------------|
| Module Configuration Register (MCR)          | S           | Yes                    | Yes                    |
| Control 1 register (CTRL1)                   | S/U         | Yes                    | No                     |
| Free Running Timer register (TIMER)          | S/U         | Yes                    | Yes                    |
| Rx Mailboxes Global Mask register (RXMGMASK) | S/U         | No                     | No                     |
| Rx Buffer 14 Mask register (RX14MASK)        | S/U         | No                     | No                     |
| Rx Buffer 15 Mask register (RX15MASK)        | S/U         | No                     | No                     |
| Error Counter Register (ECR)                 | S/U         | Yes                    | Yes                    |
| Error and Status 1 Register (ESR1)           | S/U         | Yes                    | Yes                    |

*Table continues on the next page...*

**Table 37-2. Register access and reset information (continued)**

| Register                                          | Access type | Affected by hard reset | Affected by soft reset |
|---------------------------------------------------|-------------|------------------------|------------------------|
| Interrupt Masks 2 register (IMASK2)               | S/U         | Yes                    | Yes                    |
| Interrupt Masks 1 register (IMASK1)               | S/U         | Yes                    | Yes                    |
| Interrupt Flags 2 register (IFLAG2)               | S/U         | Yes                    | Yes                    |
| Interrupt Flags 1 register (IFLAG1)               | S/U         | Yes                    | Yes                    |
| Control 2 Register (CTRL2)                        | S/U         | Yes                    | No                     |
| Error and Status 2 Register (ESR2)                | S/U         | Yes                    | Yes                    |
| CRC Register (CRCR)                               | S/U         | Yes                    | Yes                    |
| Rx FIFO Global Mask register (RXFGMASK)           | S/U         | No                     | No                     |
| Rx FIFO Information Register (RXFIR)              | S/U         | No                     | No                     |
| Message buffers                                   | S/U         | No                     | No                     |
| Rx Individual Mask Registers                      | S/U         | No                     | No                     |
| Memory Error Control Register (MECR)              | S/U         | Yes                    | Yes                    |
| Error Injection Address Register (ERRIAR)         | S/U         | Yes                    | Yes                    |
| Error Injection Data Pattern Register (ERRIDPR)   | S/U         | Yes                    | Yes                    |
| Error Injection Parity Pattern Register (ERRIPPR) | S/U         | Yes                    | Yes                    |
| Error Report Address Register (RERRAR)            | S/U         | Yes                    | Yes                    |
| Error Report Data Register (RERRDR)               | S/U         | Yes                    | Yes                    |
| Error Report Syndrome Register (RERRSYNR)         | S/U         | Yes                    | Yes                    |
| Error Status Register (ERRSR)                     | S/U         | Yes                    | Yes                    |

The FlexCAN module can store CAN messages for transmission and reception using mailboxes and Rx FIFO structures.

This module's memory map includes sixty-four 128-bit message buffers (MBs) that occupy the range from offset 0x80 to 0x47F .

### CAN memory map

| Address offset (hex) | Register name                                    | Width (in bits) | Access | Reset value | Section/ page               |
|----------------------|--------------------------------------------------|-----------------|--------|-------------|-----------------------------|
| 0                    | Module Configuration Register (CAN_MCR)          | 32              | R/W    | 5980_000Fh  | <a href="#">37.4.2/1679</a> |
| 4                    | Control 1 register (CAN_CTRL1)                   | 32              | R/W    | 0000_0000h  | <a href="#">37.4.3/1683</a> |
| 8                    | Free Running Timer (CAN_TIMER)                   | 32              | R/W    | 0000_0000h  | <a href="#">37.4.4/1686</a> |
| 10                   | Rx Mailboxes Global Mask Register (CAN_RXMGMASK) | 32              | R/W    | FFFF_FFFFh  | <a href="#">37.4.5/1687</a> |
| 14                   | Rx 14 Mask register (CAN_RX14MASK)               | 32              | R/W    | FFFF_FFFFh  | <a href="#">37.4.6/1688</a> |
| 18                   | Rx 15 Mask register (CAN_RX15MASK)               | 32              | R/W    | FFFF_FFFFh  | <a href="#">37.4.7/1689</a> |
| 1C                   | Error Counter (CAN_ECR)                          | 32              | R/W    | 0000_0000h  | <a href="#">37.4.8/1690</a> |
| 20                   | Error and Status 1 register (CAN_ESR1)           | 32              | R/W    | 0000_0000h  | <a href="#">37.4.9/1691</a> |

*Table continues on the next page...*

## CAN memory map (continued)

| Address offset (hex) | Register name                               | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|---------------------------------------------|-----------------|--------|-------------|------------------------------|
| 24                   | Interrupt Masks 2 register (CAN_IMASK2)     | 32              | R/W    | 0000_0000h  | <a href="#">37.4.10/1695</a> |
| 28                   | Interrupt Masks 1 register (CAN_IMASK1)     | 32              | R/W    | 0000_0000h  | <a href="#">37.4.11/1696</a> |
| 2C                   | Interrupt Flags 2 register (CAN_IFLAG2)     | 32              | R/W    | 0000_0000h  | <a href="#">37.4.12/1696</a> |
| 30                   | Interrupt Flags 1 register (CAN_IFLAG1)     | 32              | R/W    | 0000_0000h  | <a href="#">37.4.13/1697</a> |
| 34                   | Control 2 register (CAN_CTRL2)              | 32              | R/W    | 0080_0000h  | <a href="#">37.4.14/1699</a> |
| 38                   | Error and Status 2 register (CAN_ESR2)      | 32              | R/W    | 0000_0000h  | <a href="#">37.4.15/1703</a> |
| 44                   | CRC Register (CAN_CRCR)                     | 32              | R      | 0000_0000h  | <a href="#">37.4.16/1704</a> |
| 48                   | Rx FIFO Global Mask register (CAN_RXFGMASK) | 32              | R/W    | FFFF_FFFFh  | <a href="#">37.4.17/1705</a> |
| 4C                   | Rx FIFO Information Register (CAN_RXFIR)    | 32              | R      | Undefined   | <a href="#">37.4.18/1706</a> |
| 880                  | Rx Individual Mask Registers (CAN_RXIMR0)   | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 884                  | Rx Individual Mask Registers (CAN_RXIMR1)   | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 888                  | Rx Individual Mask Registers (CAN_RXIMR2)   | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 88C                  | Rx Individual Mask Registers (CAN_RXIMR3)   | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 890                  | Rx Individual Mask Registers (CAN_RXIMR4)   | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 894                  | Rx Individual Mask Registers (CAN_RXIMR5)   | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 898                  | Rx Individual Mask Registers (CAN_RXIMR6)   | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 89C                  | Rx Individual Mask Registers (CAN_RXIMR7)   | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 8A0                  | Rx Individual Mask Registers (CAN_RXIMR8)   | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 8A4                  | Rx Individual Mask Registers (CAN_RXIMR9)   | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 8A8                  | Rx Individual Mask Registers (CAN_RXIMR10)  | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 8AC                  | Rx Individual Mask Registers (CAN_RXIMR11)  | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 8B0                  | Rx Individual Mask Registers (CAN_RXIMR12)  | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |

Table continues on the next page...

## CAN memory map (continued)

| Address offset (hex) | Register name                              | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|--------------------------------------------|-----------------|--------|-------------|------------------------------|
| 8B4                  | Rx Individual Mask Registers (CAN_RXIMR13) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 8B8                  | Rx Individual Mask Registers (CAN_RXIMR14) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 8BC                  | Rx Individual Mask Registers (CAN_RXIMR15) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 8C0                  | Rx Individual Mask Registers (CAN_RXIMR16) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 8C4                  | Rx Individual Mask Registers (CAN_RXIMR17) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 8C8                  | Rx Individual Mask Registers (CAN_RXIMR18) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 8CC                  | Rx Individual Mask Registers (CAN_RXIMR19) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 8D0                  | Rx Individual Mask Registers (CAN_RXIMR20) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 8D4                  | Rx Individual Mask Registers (CAN_RXIMR21) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 8D8                  | Rx Individual Mask Registers (CAN_RXIMR22) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 8DC                  | Rx Individual Mask Registers (CAN_RXIMR23) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 8E0                  | Rx Individual Mask Registers (CAN_RXIMR24) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 8E4                  | Rx Individual Mask Registers (CAN_RXIMR25) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 8E8                  | Rx Individual Mask Registers (CAN_RXIMR26) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 8EC                  | Rx Individual Mask Registers (CAN_RXIMR27) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 8F0                  | Rx Individual Mask Registers (CAN_RXIMR28) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 8F4                  | Rx Individual Mask Registers (CAN_RXIMR29) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 8F8                  | Rx Individual Mask Registers (CAN_RXIMR30) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 8FC                  | Rx Individual Mask Registers (CAN_RXIMR31) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 900                  | Rx Individual Mask Registers (CAN_RXIMR32) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 904                  | Rx Individual Mask Registers (CAN_RXIMR33) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 908                  | Rx Individual Mask Registers (CAN_RXIMR34) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |

Table continues on the next page...



## CAN memory map (continued)

| Address offset (hex) | Register name                              | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|--------------------------------------------|-----------------|--------|-------------|------------------------------|
| 90C                  | Rx Individual Mask Registers (CAN_RXIMR35) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 910                  | Rx Individual Mask Registers (CAN_RXIMR36) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 914                  | Rx Individual Mask Registers (CAN_RXIMR37) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 918                  | Rx Individual Mask Registers (CAN_RXIMR38) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 91C                  | Rx Individual Mask Registers (CAN_RXIMR39) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 920                  | Rx Individual Mask Registers (CAN_RXIMR40) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 924                  | Rx Individual Mask Registers (CAN_RXIMR41) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 928                  | Rx Individual Mask Registers (CAN_RXIMR42) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 92C                  | Rx Individual Mask Registers (CAN_RXIMR43) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 930                  | Rx Individual Mask Registers (CAN_RXIMR44) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 934                  | Rx Individual Mask Registers (CAN_RXIMR45) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 938                  | Rx Individual Mask Registers (CAN_RXIMR46) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 93C                  | Rx Individual Mask Registers (CAN_RXIMR47) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 940                  | Rx Individual Mask Registers (CAN_RXIMR48) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 944                  | Rx Individual Mask Registers (CAN_RXIMR49) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 948                  | Rx Individual Mask Registers (CAN_RXIMR50) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 94C                  | Rx Individual Mask Registers (CAN_RXIMR51) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 950                  | Rx Individual Mask Registers (CAN_RXIMR52) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 954                  | Rx Individual Mask Registers (CAN_RXIMR53) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 958                  | Rx Individual Mask Registers (CAN_RXIMR54) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 95C                  | Rx Individual Mask Registers (CAN_RXIMR55) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 960                  | Rx Individual Mask Registers (CAN_RXIMR56) | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |

Table continues on the next page...

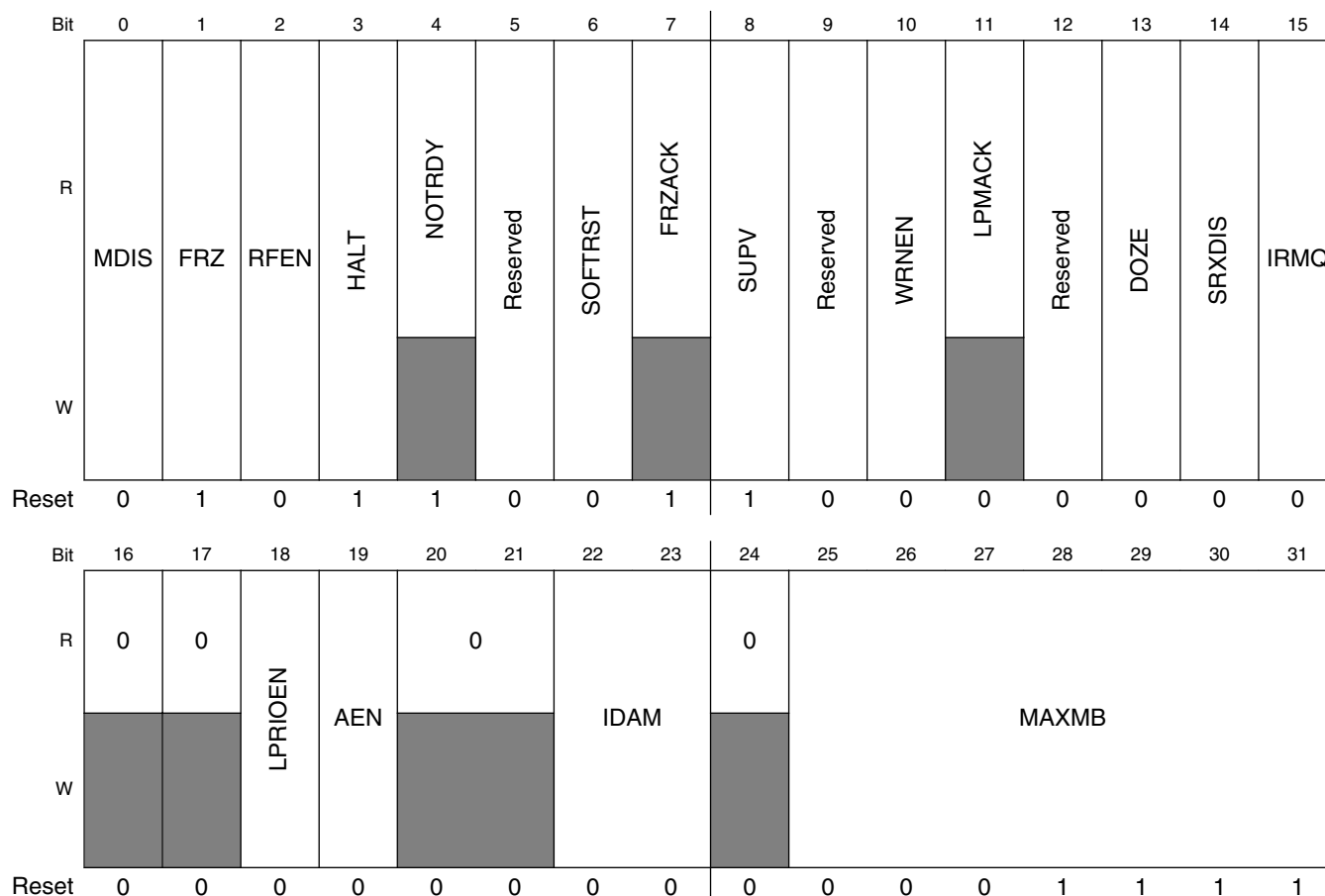
## CAN memory map (continued)

| Address offset (hex) | Register name                                         | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|-------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 964                  | Rx Individual Mask Registers (CAN_RXIMR57)            | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 968                  | Rx Individual Mask Registers (CAN_RXIMR58)            | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 96C                  | Rx Individual Mask Registers (CAN_RXIMR59)            | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 970                  | Rx Individual Mask Registers (CAN_RXIMR60)            | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 974                  | Rx Individual Mask Registers (CAN_RXIMR61)            | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 978                  | Rx Individual Mask Registers (CAN_RXIMR62)            | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| 97C                  | Rx Individual Mask Registers (CAN_RXIMR63)            | 32              | R/W    | Undefined   | <a href="#">37.4.19/1706</a> |
| AE0                  | Memory Error Control Register (CAN_MECCR)             | 32              | R/W    | 800C_0080h  | <a href="#">37.4.20/1707</a> |
| AE4                  | Error Injection Address Register (CAN_ERRIAR)         | 32              | R/W    | 0000_0000h  | <a href="#">37.4.21/1709</a> |
| AE8                  | Error Injection Data Pattern Register (CAN_ERRIDPR)   | 32              | R/W    | 0000_0000h  | <a href="#">37.4.22/1711</a> |
| AEC                  | Error Injection Parity Pattern Register (CAN_ERRIPPR) | 32              | R/W    | 0000_0000h  | <a href="#">37.4.23/1711</a> |
| AF0                  | Error Report Address Register (CAN_RERRAR)            | 32              | R      | 0000_0000h  | <a href="#">37.4.24/1712</a> |
| AF4                  | Error Report Data Register (CAN_RERRDR)               | 32              | R      | 0000_0000h  | <a href="#">37.4.25/1713</a> |
| AF8                  | Error Report Syndrome Register (CAN_RERRSYNR)         | 32              | R      | 0000_0000h  | <a href="#">37.4.26/1714</a> |
| AFC                  | Error Status Register (CAN_ERRSR)                     | 32              | R/W    | 0000_0000h  | <a href="#">37.4.27/1716</a> |

## 37.4.2 Module Configuration Register (CAN\_MCR)

This register defines global system configurations, such as the module operation modes and the maximum message buffer configuration.

Address: 0h base + 0h offset = 0h



**CAN\_MCR field descriptions**

| Field     | Description                                                                                                                                                                                                                                                                                                                                    |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>MDIS | <p>Module Disable</p> <p>This bit controls whether FlexCAN is enabled or not. When disabled, FlexCAN disables the clocks to the CAN Protocol Engine and Controller Host Interface sub-modules. This is the only bit within this register not affected by soft reset.</p> <p>0 Enable the FlexCAN module.<br/>1 Disable the FlexCAN module.</p> |
| 1<br>FRZ  | <p>Freeze Enable</p> <p>The FRZ bit specifies the FlexCAN behavior when the HALT bit in the MCR Register is set or when Debug mode is requested at chip level . When FRZ is asserted, FlexCAN is enabled to enter Freeze mode. Negation of this bit field causes FlexCAN to exit from Freeze mode.</p>                                         |

*Table continues on the next page...*

**CAN\_MCR field descriptions (continued)**

| Field         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               | 0 Not enabled to enter Freeze mode.<br>1 Enabled to enter Freeze mode.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 2<br>RFEN     | <b>Rx FIFO Enable</b><br><br>This bit controls whether the Rx FIFO feature is enabled or not. When RFEN is set, MBs 0 to 5 cannot be used for normal reception and transmission because the corresponding memory region (0x80-0xDC) is used by the FIFO engine as well as additional MBs (up to 32, depending on CTRL2[RFFN] setting) which are used as Rx FIFO ID Filter Table elements. RFEN also impacts the definition of the minimum number of peripheral clocks per CAN bit as described in the table "Minimum Ratio Between Peripheral Clock Frequency and CAN Bit Rate" (in section "Arbitration and Matching Timing"). This bit can be written only in Freeze mode because it is blocked by hardware in other modes.<br><br>0 Rx FIFO not enabled.<br>1 Rx FIFO enabled.                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 3<br>HALT     | <b>Halt FlexCAN</b><br><br>Assertion of this bit puts the FlexCAN module into Freeze mode. The CPU should clear it after initializing the Message Buffers and Control Register. No reception or transmission is performed by FlexCAN before this bit is cleared. Freeze mode cannot be entered while FlexCAN is in a low power mode.<br><br>0 No Freeze mode request.<br>1 Enters Freeze mode if the FRZ bit is asserted.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 4<br>NOTRDY   | <b>FlexCAN Not Ready</b><br><br>This read-only bit indicates that FlexCAN is either in Disable mode , Doze mode, Stop mode or Freeze mode. It is negated once FlexCAN has exited these modes.<br><br>0 FlexCAN module is either in Normal mode, Listen-Only mode or Loop-Back mode.<br>1 FlexCAN module is either in Disable mode , Doze mode, Stop mode or Freeze mode.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 5<br>Reserved | This field is reserved.<br>Always write 0 to this field.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 6<br>SOFTTRST | <b>Soft Reset</b><br><br>When this bit is asserted, FlexCAN resets its internal state machines and some of the memory mapped registers. The following registers are reset: MCR (except the MDIS bit), TIMER , ECR, ESR1, ESR2, IMASK1, IMASK2, IFLAG1, IFLAG2 and CRCR. Configuration registers that control the interface to the CAN bus are not affected by soft reset. The following registers are unaffected: CTRL1, CTRL2, all RXIMR registers, RXMGMASK, RX14MASK, RX15MASK, RXFGMASK, RXFIR, all Message Buffers .<br><br>The SOFTTRST bit can be asserted directly by the CPU when it writes to the MCR Register. Because soft reset is synchronous and has to follow a request/acknowledge procedure across clock domains, it may take some time to fully propagate its effect. The SOFTTRST bit remains asserted while reset is pending, and is automatically negated when reset completes. Therefore, software can poll this bit to know when the soft reset has completed.<br><br>Soft reset cannot be applied while clocks are shut down in a low power mode. The module should be first removed from low power mode, and then soft reset can be applied.<br><br>0 No reset request.<br>1 Resets the registers affected by soft reset. |
| 7<br>FRZACK   | <b>Freeze Mode Acknowledge</b><br><br>This read-only bit indicates that FlexCAN is in Freeze mode and its prescaler is stopped. The Freeze mode request cannot be granted until current transmission or reception processes have finished.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

*Table continues on the next page...*

## CAN\_MCR field descriptions (continued)

| Field          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | <p>Therefore the software can poll the FRZACK bit to know when FlexCAN has actually entered Freeze mode. If Freeze Mode request is negated, then this bit is negated after the FlexCAN prescaler is running again. If Freeze mode is requested while FlexCAN is in a low power mode, then the FRZACK bit will be set only when the low-power mode is exited. See Section "Freeze Mode".</p> <p><b>NOTE:</b> FRZACK will be asserted within 178 CAN bits from the freeze mode request by the CPU, and negated within 2 CAN bits after the freeze mode request removal (see Section "Protocol Timing").</p> <p>0 FlexCAN not in Freeze mode, prescaler running.<br/>1 FlexCAN in Freeze mode, prescaler stopped.</p> |
| 8<br>SUPV      | <p>Supervisor Mode</p> <p>This bit configures the FlexCAN to be either in Supervisor or User mode. The registers affected by this bit are marked as S/U in the Access Type column of the module memory map. Reset value of this bit is 1, so the affected registers start with Supervisor access allowance only. This bit can be written only in Freeze mode because it is blocked by hardware in other modes.</p> <p>0 FlexCAN is in User mode. Affected registers allow both Supervisor and Unrestricted accesses.<br/>1 FlexCAN is in Supervisor mode. Affected registers allow only Supervisor access. Unrestricted access behaves as though the access was done to an unimplemented register location.</p>    |
| 9<br>Reserved  | <p>This field is reserved.<br/>Always write 0 to this field.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 10<br>WRNEN    | <p>Warning Interrupt Enable</p> <p>When asserted, this bit enables the generation of the TWRNINT and RWRNINT flags in the Error and Status Register. If WRNEN is negated, the TWRNINT and RWRNINT flags will always be zero, independent of the values of the error counters, and no warning interrupt will ever be generated. This bit can be written only in Freeze mode because it is blocked by hardware in other modes.</p> <p>0 TWRNINT and RWRNINT bits are zero, independent of the values in the error counters.<br/>1 TWRNINT and RWRNINT bits are set when the respective error counter transitions from less than 96 to greater than or equal to 96.</p>                                               |
| 11<br>LPMACK   | <p>Low-Power Mode Acknowledge</p> <p>This read-only bit indicates that FlexCAN is in a low-power mode (Disable mode, Doze mode, Stop mode). A low-power mode cannot be entered until all current transmission or reception processes have finished, so the CPU can poll the LPMACK bit to know when FlexCAN has actually entered low power mode.</p> <p><b>NOTE:</b> LPMACK will be asserted within 180 CAN bits from the low-power mode request by the CPU, and negated within 2 CAN bits after the low-power mode request removal (see Section "Protocol Timing").</p> <p>0 FlexCAN is not in a low-power mode.<br/>1 FlexCAN is in a low-power mode.</p>                                                        |
| 12<br>Reserved | <p>This field is reserved.<br/>Always write 0 to this field.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 13<br>DOZE     | <p>Doze Mode Enable</p> <p>This bit defines whether FlexCAN is allowed to enter low-power mode when Doze mode is requested at chip level.</p> <p>0 FlexCAN is not enabled to enter low-power mode when Doze mode is requested.<br/>1 FlexCAN is enabled to enter low-power mode when Doze mode is requested.</p>                                                                                                                                                                                                                                                                                                                                                                                                   |

Table continues on the next page...

## CAN\_MCR field descriptions (continued)

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 14<br>SRXDIS      | <p>Self Reception Disable</p> <p>This bit defines whether FlexCAN is allowed to receive frames transmitted by itself. If this bit is asserted, frames transmitted by the module will not be stored in any MB, regardless if the MB is programmed with an ID that matches the transmitted frame, and no interrupt flag or interrupt signal will be generated due to the frame reception. This bit can be written only in Freeze mode because it is blocked by hardware in other modes.</p> <p>0 Self reception enabled.<br/>1 Self reception disabled.</p>                                                                                                                                                                                                              |
| 15<br>IRMQ        | <p>Individual Rx Masking And Queue Enable</p> <p>This bit indicates whether Rx matching process will be based either on individual masking and queue or on masking scheme with RXMGMASK, RX14MASK and RX15MASK, RXFGMASK. This bit can be written only in Freeze mode because it is blocked by hardware in other modes.</p> <p>0 Individual Rx masking and queue feature are disabled. For backward compatibility with legacy applications, the reading of C/S word locks the MB even if it is EMPTY.<br/>1 Individual Rx masking and queue feature are enabled.</p>                                                                                                                                                                                                   |
| 16<br>Reserved    | <p>This field is reserved.<br/>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 17<br>Reserved    | <p>This field is reserved.<br/>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 18<br>LPRIOEN     | <p>Local Priority Enable</p> <p>This bit is provided for backwards compatibility with legacy applications. It controls whether the local priority feature is enabled or not. It is used to expand the ID used during the arbitration process. With this expanded ID concept, the arbitration process is done based on the full 32-bit word, but the actual transmitted ID still has 11-bit for standard frames and 29-bit for extended frames. This bit can be written only in Freeze mode because it is blocked by hardware in other modes.</p> <p>0 Local Priority disabled.<br/>1 Local Priority enabled.</p>                                                                                                                                                       |
| 19<br>AEN         | <p>Abort Enable</p> <p>This bit is supplied for backwards compatibility with legacy applications. When asserted, it enables the Tx abort mechanism. This mechanism guarantees a safe procedure for aborting a pending transmission, so that no frame is sent in the CAN bus without notification. This bit can be written only in Freeze mode because it is blocked by hardware in other modes.</p> <p><b>NOTE:</b> When MCR[AEN] is asserted, only the abort mechanism (see Section "Transmission Abort Mechanism") must be used for updating Mailboxes configured for transmission.</p> <p><b>CAUTION:</b> Writing the Abort code into Rx Mailboxes can cause unpredictable results when the MCR[AEN] is asserted.</p> <p>0 Abort disabled.<br/>1 Abort enabled.</p> |
| 20–21<br>Reserved | <p>This field is reserved.<br/>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 22–23<br>IDAM     | ID Acceptance Mode                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

Table continues on the next page...

## CAN\_MCR field descriptions (continued)

| Field          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | <p>This 2-bit field identifies the format of the Rx FIFO ID Filter Table elements. Note that all elements of the table are configured at the same time by this field (they are all the same format). See Section "Rx FIFO Structure". This field can be written only in Freeze mode because it is blocked by hardware in other modes.</p> <p>00 Format A: One full ID (standard and extended) per ID Filter Table element.</p> <p>01 Format B: Two full standard IDs or two partial 14-bit (standard and extended) IDs per ID Filter Table element.</p> <p>10 Format C: Four partial 8-bit Standard IDs per ID Filter Table element.</p> <p>11 Format D: All frames rejected.</p>                                                                                                                                                                                                                           |
| 24<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 25–31<br>MAXMB | <p>Number Of The Last Message Buffer</p> <p>This 7-bit field defines the number of the last Message Buffers that will take part in the matching and arbitration processes. The reset value (0x0F) is equivalent to a 16 MB configuration. This field can be written only in Freeze mode because it is blocked by hardware in other modes.</p> <p>Number of the last MB = MAXMB</p> <p><b>NOTE:</b> MAXMB must be programmed with a value smaller than the parameter NUMBER_OF_MB, otherwise the number of the last effective Message Buffer will be: (NUMBER_OF_MB - 1)</p> <p>Additionally, the value of MAXMB must encompass the FIFO size defined by CTRL2[RFFN]. MAXMB also impacts the definition of the minimum number of peripheral clocks per CAN bit as described in Table "Minimum Ratio Between Peripheral Clock Frequency and CAN Bit Rate" (in Section "Arbitration and Matching Timing").</p> |

## 37.4.3 Control 1 register (CAN\_CTRL1)

This register is defined for specific FlexCAN control features related to the CAN bus, such as bit-rate, programmable sampling point within an Rx bit, Loop Back mode, Listen-Only mode, Bus Off recovery behavior and interrupt enabling (Bus-Off, Error, Warning). It also determines the Division Factor for the clock prescaler.

Address: 0h base + 4h offset = 4h

|       |         |        |        |     |         |         |    |    |     |         |      |      |       |         |    |    |
|-------|---------|--------|--------|-----|---------|---------|----|----|-----|---------|------|------|-------|---------|----|----|
| Bit   | 0       | 1      | 2      | 3   | 4       | 5       | 6  | 7  | 8   | 9       | 10   | 11   | 12    | 13      | 14 | 15 |
| R     | PRESDIV |        |        |     |         |         |    |    | RJW | PSEG1   |      |      | PSEG2 |         |    |    |
| W     |         |        |        |     |         |         |    |    |     |         |      |      |       |         |    |    |
| Reset | 0       | 0      | 0      | 0   | 0       | 0       | 0  | 0  | 0   | 0       | 0    | 0    | 0     | 0       | 0  | 0  |
| Bit   | 16      | 17     | 18     | 19  | 20      | 21      | 22 | 23 | 24  | 25      | 26   | 27   | 28    | 29      | 30 | 31 |
| R     | BOFFMSK | ERRMSK | CLKSRC | LPB | TWRNMSK | RWRNMSK | 0  |    | SMP | BOFFREC | TSYN | LBUF | LOM   | PROPSEG |    |    |
| W     |         |        |        |     |         |         |    |    |     |         |      |      |       |         |    |    |
| Reset | 0       | 0      | 0      | 0   | 0       | 0       | 0  | 0  | 0   | 0       | 0    | 0    | 0     | 0       | 0  | 0  |

## CAN\_CTRL1 field descriptions

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–7<br>PRES DIV | <p>Prescaler Division Factor</p> <p>This 8-bit field defines the ratio between the PE clock frequency and the Serial Clock (Sclck) frequency. The Sclck period defines the time quantum of the CAN protocol. For the reset value, the Sclck frequency is equal to the PE clock frequency. The Maximum value of this field is 0xFF, that gives a minimum Sclck frequency equal to the PE clock frequency divided by 256. See Section "Protocol Timing". This field can be written only in Freeze mode because it is blocked by hardware in other modes.</p> <p><math>\text{Sclck frequency} = \text{PE clock frequency} / (\text{PRES DIV} + 1)</math></p>                       |
| 8–9<br>RJW      | <p>Resync Jump Width</p> <p>This 2-bit field defines the maximum number of time quanta that a bit time can be changed by one re-synchronization. One time quantum is equal to the Sclck period. The valid programmable values are 0–3. This field can be written only in Freeze mode because it is blocked by hardware in other modes.</p> <p><math>\text{Resync Jump Width} = \text{RJW} + 1.</math></p>                                                                                                                                                                                                                                                                       |
| 10–12<br>PSEG1  | <p>Phase Segment 1</p> <p>This 3-bit field defines the length of Phase Buffer Segment 1 in the bit time. The valid programmable values are 0–7. This field can be written only in Freeze mode because it is blocked by hardware in other modes.</p> <p><math>\text{Phase Buffer Segment 1} = (\text{PSEG1} + 1) \times \text{Time-Quanta}.</math></p>                                                                                                                                                                                                                                                                                                                           |
| 13–15<br>PSEG2  | <p>Phase Segment 2</p> <p>This 3-bit field defines the length of Phase Buffer Segment 2 in the bit time. The valid programmable values are 1–7. This field can be written only in Freeze mode because it is blocked by hardware in other modes.</p> <p><math>\text{Phase Buffer Segment 2} = (\text{PSEG2} + 1) \times \text{Time-Quanta}.</math></p>                                                                                                                                                                                                                                                                                                                           |
| 16<br>BOFFMSK   | <p>Bus Off Mask</p> <p>This bit provides a mask for the Bus Off Interrupt.</p> <p>0 Bus Off interrupt disabled.<br/>1 Bus Off interrupt enabled.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 17<br>ERRMSK    | <p>Error Mask</p> <p>This bit provides a mask for the Error Interrupt.</p> <p>0 Error interrupt disabled.<br/>1 Error interrupt enabled.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 18<br>CLKSRC    | <p>CAN Engine Clock Source</p> <p>This bit selects the clock source to the CAN Protocol Engine (PE) to be either the peripheral clock (driven by the PLL) or the crystal oscillator clock. The selected clock is the one fed to the prescaler to generate the Serial Clock (Sclck). In order to guarantee reliable operation, this bit can be written only in Disable mode because it is blocked by hardware in other modes. See Section "Protocol Timing".</p> <p>0 The CAN engine clock source is the oscillator clock. Under this condition, the oscillator clock frequency must be lower than the bus clock.<br/>1 The CAN engine clock source is the peripheral clock.</p> |
| 19<br>LPB       | <p>Loop Back Mode</p> <p>This bit configures FlexCAN to operate in Loop-Back mode. In this mode, FlexCAN performs an internal loop back that can be used for self test operation. The bit stream output of the transmitter is fed back</p>                                                                                                                                                                                                                                                                                                                                                                                                                                      |

*Table continues on the next page...*



## CAN\_CTRL1 field descriptions (continued)

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | <p>internally to the receiver input. The Rx CAN input pin is ignored and the Tx CAN output goes to the recessive state (logic 1). FlexCAN behaves as it normally does when transmitting, and treats its own transmitted message as a message received from a remote node. In this mode, FlexCAN ignores the bit sent during the ACK slot in the CAN frame acknowledge field, generating an internal acknowledge bit to ensure proper reception of its own message. Both transmit and receive interrupts are generated. This bit can be written only in Freeze mode because it is blocked by hardware in other modes.</p> <p><b>NOTE:</b> In this mode, the MCR[SRXDIS] cannot be asserted because this will impede the self reception of a transmitted message.</p> <p>0 Loop Back disabled.<br/>1 Loop Back enabled.</p>                                                                                                                                                                                                                                                                                                                                              |
| 20<br>TWRNMSK     | <p>Tx Warning Interrupt Mask</p> <p>This bit provides a mask for the Tx Warning Interrupt associated with the TWRNINT flag in the Error and Status Register. This bit is read as zero when MCR[WRNEN] bit is negated. This bit can be written only if MCR[WRNEN] bit is asserted.</p> <p>0 Tx Warning Interrupt disabled.<br/>1 Tx Warning Interrupt enabled.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 21<br>RWRNMSK     | <p>Rx Warning Interrupt Mask</p> <p>This bit provides a mask for the Rx Warning Interrupt associated with the RWRNINT flag in the Error and Status Register. This bit is read as zero when MCR[WRNEN] bit is negated. This bit can be written only if MCR[WRNEN] bit is asserted.</p> <p>0 Rx Warning Interrupt disabled.<br/>1 Rx Warning Interrupt enabled.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 22–23<br>Reserved | <p>This field is reserved.<br/>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 24<br>SMP         | <p>CAN Bit Sampling</p> <p>This bit defines the sampling mode of CAN bits at the Rx input. This bit can be written only in Freeze mode because it is blocked by hardware in other modes.</p> <p>0 Just one sample is used to determine the bit value.<br/>1 Three samples are used to determine the value of the received bit: the regular one (sample point) and 2 preceding samples; a majority rule is used.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 25<br>BOFFREC     | <p>Bus Off Recovery</p> <p>This bit defines how FlexCAN recovers from Bus Off state. If this bit is negated, automatic recovering from Bus Off state occurs according to the CAN Specification 2.0B. If the bit is asserted, automatic recovering from Bus Off is disabled and the module remains in Bus Off state until the bit is negated by the user. If the negation occurs before 128 sequences of 11 recessive bits are detected on the CAN bus, then Bus Off recovery happens as if the BOFFREC bit had never been asserted. If the negation occurs after 128 sequences of 11 recessive bits occurred, then FlexCAN will re-synchronize to the bus by waiting for 11 recessive bits before joining the bus. After negation, the BOFFREC bit can be re-asserted again during Bus Off, but it will be effective only the next time the module enters Bus Off. If BOFFREC was negated when the module entered Bus Off, asserting it during Bus Off will not be effective for the current Bus Off recovery.</p> <p>0 Automatic recovering from Bus Off state enabled, according to CAN Spec 2.0 part B.<br/>1 Automatic recovering from Bus Off state disabled.</p> |

Table continues on the next page...

**CAN\_CTRL1 field descriptions (continued)**

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 26<br>TSYN       | <p>Timer Sync</p> <p>This bit enables a mechanism that resets the free-running timer each time a message is received in Message Buffer 0. This feature provides means to synchronize multiple FlexCAN stations with a special “SYNC” message, that is, global network time. If the RFEN bit in MCR is set (Rx FIFO enabled), the first available Mailbox, according to CTRL2[RFFN] setting, is used for timer synchronization instead of MB0. This bit can be written only in Freeze mode because it is blocked by hardware in other modes.</p> <p>0 Timer Sync feature disabled<br/>1 Timer Sync feature enabled</p>                                                                                                                                                                                                                                                                          |
| 27<br>LBUF       | <p>Lowest Buffer Transmitted First</p> <p>This bit defines the ordering mechanism for Message Buffer transmission. When asserted, the LPRIEN bit does not affect the priority arbitration. This bit can be written only in Freeze mode because it is blocked by hardware in other modes.</p> <p>0 Buffer with highest priority is transmitted first.<br/>1 Lowest number buffer is transmitted first.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 28<br>LOM        | <p>Listen-Only Mode</p> <p>This bit configures FlexCAN to operate in Listen-Only mode. In this mode, transmission is disabled, all error counters are frozen and the module operates in a CAN Error Passive mode. Only messages acknowledged by another CAN station will be received. If FlexCAN detects a message that has not been acknowledged, it will flag a BIT0 error without changing the REC, as if it was trying to acknowledge the message.</p> <p>Listen-Only mode acknowledgement can be obtained by the state of ESR1[FLTCNF] field which is Passive Error when Listen-Only mode is entered. There can be some delay between the Listen-Only mode request and acknowledge.</p> <p>This bit can be written only in Freeze mode because it is blocked by hardware in other modes.</p> <p>0 Listen-Only mode is deactivated.<br/>1 FlexCAN module operates in Listen-Only mode.</p> |
| 29–31<br>PROPSEG | <p>Propagation Segment</p> <p>This 3-bit field defines the length of the Propagation Segment in the bit time. The valid programmable values are 0–7. This field can be written only in Freeze mode because it is blocked by hardware in other modes.</p> <p>Propagation Segment Time = (PROPSEG + 1) × Time-Quanta.<br/>Time-Quantum = one Sclock period.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

**37.4.4 Free Running Timer (CAN\_TIMER)**

This register represents a 16-bit free running counter that can be read and written by the CPU. The timer starts from 0x0 after Reset, counts linearly to 0xFFFF, and wraps around.

The timer is clocked by the FlexCAN bit-clock, which defines the baud rate on the CAN bus. During a message transmission/reception, it increments by one for each bit that is received or transmitted. When there is no message on the bus, it counts using the previously programmed baud rate. The timer is not incremented during Disable, Doze, Stop, and Freeze modes.

The timer value is captured when the second bit of the identifier field of any frame is on the CAN bus. This captured value is written into the Time Stamp entry in a message buffer after a successful reception or transmission of a message.

If bit CTRL1[TSYN] is asserted, the Timer is reset whenever a message is received in the first available Mailbox, according to CTRL2[RFFN] setting.

The CPU can write to this register anytime. However, if the write occurs at the same time that the Timer is being reset by a reception in the first Mailbox, then the write value is discarded.

Reading this register affects the Mailbox Unlocking procedure; see Section "Mailbox Lock Mechanism".

Address: 0h base + 8h offset = 8h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16    | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | TIMER |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0     | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**CAN\_TIMER field descriptions**

| Field            | Description                                                                             |
|------------------|-----------------------------------------------------------------------------------------|
| 0–15<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 16–31<br>TIMER   | Timer Value<br>Contains the free-running counter value.                                 |

### 37.4.5 Rx Mailboxes Global Mask Register (CAN\_RXMGMASK)

This register is located in RAM.

RXMGMASK is provided for legacy application support.

- When the MCR[IRMQ] bit is negated, RXMGMASK is always in effect.
- When the MCR[IRMQ] bit is asserted, RXMGMASK has no effect.

RXMGMASK is used to mask the filter fields of all Rx MBs, excluding MBs 14-15, which have individual mask registers.

This register can only be written in Freeze mode as it is blocked by hardware in other modes.

Address: 0h base + 10h offset = 10h

|       |          |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0        | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |          |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |          |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|       | MG[31:0] |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 1        | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  |    |

### CAN\_RXMGMASK field descriptions

| Field                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                       |                   |                   |                       |                       |  |  |         |         |        |          |   |   |   |                   |                   |          |           |   |   |   |        |        |          |        |   |   |   |   |   |   |          |   |   |   |   |   |          |           |   |   |   |        |        |          |        |
|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|-------------------|-------------------|-----------------------|-----------------------|--|--|---------|---------|--------|----------|---|---|---|-------------------|-------------------|----------|-----------|---|---|---|--------|--------|----------|--------|---|---|---|---|---|---|----------|---|---|---|---|---|----------|-----------|---|---|---|--------|--------|----------|--------|
| 0–31<br>MG[31:0]      | <p>Rx Mailboxes Global Mask Bits</p> <p>These bits mask the Mailbox filter bits. Note that the alignment with the ID word of the Mailbox is not perfect as the two most significant MG bits affect the fields RTR and IDE, which are located in the Control and Status word of the Mailbox. The following table shows in detail which MG bits mask each Mailbox filter field.</p> <table><tr><th rowspan="2">SMB[RTR] <sup>1</sup></th><th rowspan="2">CTRL2[RRS]</th><th rowspan="2">CTRL2[EACE<br/>N]</th><th colspan="4">Mailbox filter fields</th></tr><tr><th>MB[RTR]</th><th>MB[IDE]</th><th>MB[ID]</th><th>Reserved</th></tr><tr><td>0</td><td>-</td><td>0</td><td>note <sup>2</sup></td><td>note <sup>3</sup></td><td>MG[28:0]</td><td>MG[31:29]</td></tr><tr><td>0</td><td>-</td><td>1</td><td>MG[31]</td><td>MG[30]</td><td>MG[28:0]</td><td>MG[29]</td></tr><tr><td>1</td><td>0</td><td>-</td><td>-</td><td>-</td><td>-</td><td>MG[31:0]</td></tr><tr><td>1</td><td>1</td><td>0</td><td>-</td><td>-</td><td>MG[28:0]</td><td>MG[31:29]</td></tr><tr><td>1</td><td>1</td><td>1</td><td>MG[31]</td><td>MG[30]</td><td>MG[28:0]</td><td>MG[29]</td></tr></table> <p>1. RTR bit of the Incoming Frame. It is saved into an auxiliary MB called Rx Serial Message Buffer (Rx SMB).</p> <p>2. If the CTRL2[EACEN] bit is negated, the RTR bit of Mailbox is never compared with the RTR bit of the incoming frame.</p> <p>3. If the CTRL2[EACEN] bit is negated, the IDE bit of Mailbox is always compared with the IDE bit of the incoming frame.</p> <p>0 The corresponding bit in the filter is "don't care."</p> <p>1 The corresponding bit in the filter is checked.</p> | SMB[RTR] <sup>1</sup> | CTRL2[RRS]        | CTRL2[EACE<br>N]  | Mailbox filter fields |                       |  |  | MB[RTR] | MB[IDE] | MB[ID] | Reserved | 0 | - | 0 | note <sup>2</sup> | note <sup>3</sup> | MG[28:0] | MG[31:29] | 0 | - | 1 | MG[31] | MG[30] | MG[28:0] | MG[29] | 1 | 0 | - | - | - | - | MG[31:0] | 1 | 1 | 0 | - | - | MG[28:0] | MG[31:29] | 1 | 1 | 1 | MG[31] | MG[30] | MG[28:0] | MG[29] |
| SMB[RTR] <sup>1</sup> | CTRL2[RRS]                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                       |                   |                   | CTRL2[EACE<br>N]      | Mailbox filter fields |  |  |         |         |        |          |   |   |   |                   |                   |          |           |   |   |   |        |        |          |        |   |   |   |   |   |   |          |   |   |   |   |   |          |           |   |   |   |        |        |          |        |
|                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | MB[RTR]               | MB[IDE]           | MB[ID]            |                       | Reserved              |  |  |         |         |        |          |   |   |   |                   |                   |          |           |   |   |   |        |        |          |        |   |   |   |   |   |   |          |   |   |   |   |   |          |           |   |   |   |        |        |          |        |
| 0                     | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 0                     | note <sup>2</sup> | note <sup>3</sup> | MG[28:0]              | MG[31:29]             |  |  |         |         |        |          |   |   |   |                   |                   |          |           |   |   |   |        |        |          |        |   |   |   |   |   |   |          |   |   |   |   |   |          |           |   |   |   |        |        |          |        |
| 0                     | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 1                     | MG[31]            | MG[30]            | MG[28:0]              | MG[29]                |  |  |         |         |        |          |   |   |   |                   |                   |          |           |   |   |   |        |        |          |        |   |   |   |   |   |   |          |   |   |   |   |   |          |           |   |   |   |        |        |          |        |
| 1                     | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | -                     | -                 | -                 | -                     | MG[31:0]              |  |  |         |         |        |          |   |   |   |                   |                   |          |           |   |   |   |        |        |          |        |   |   |   |   |   |   |          |   |   |   |   |   |          |           |   |   |   |        |        |          |        |
| 1                     | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 0                     | -                 | -                 | MG[28:0]              | MG[31:29]             |  |  |         |         |        |          |   |   |   |                   |                   |          |           |   |   |   |        |        |          |        |   |   |   |   |   |   |          |   |   |   |   |   |          |           |   |   |   |        |        |          |        |
| 1                     | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | 1                     | MG[31]            | MG[30]            | MG[28:0]              | MG[29]                |  |  |         |         |        |          |   |   |   |                   |                   |          |           |   |   |   |        |        |          |        |   |   |   |   |   |   |          |   |   |   |   |   |          |           |   |   |   |        |        |          |        |

1. RTR bit of the Incoming Frame. It is saved into an auxiliary MB called Rx Serial Message Buffer (Rx SMB).
2. If the CTRL2[EACEN] bit is negated, the RTR bit of Mailbox is never compared with the RTR bit of the incoming frame.
3. If the CTRL2[EACEN] bit is negated, the IDE bit of Mailbox is always compared with the IDE bit of the incoming frame.

## 37.4.6 Rx 14 Mask register (CAN\_RX14MASK)

This register is located in RAM.

RX14MASK is provided for legacy application support. When the MCR[IRMQ] bit is asserted, RX14MASK has no effect.

RX14MASK is used to mask the filter fields of Message Buffer 14.

This register can only be programmed while the module is in Freeze mode as it is blocked by hardware in other modes.

Address: 0h base + 14h offset = 14h

|       |             |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0           | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | RX14M[31:0] |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |             |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 1           | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  |    |

### CAN\_RX14MASK field descriptions

| Field               | Description                                                                                                                                                                                                                                                                                                                            |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>RX14M[31:0] | <p>Rx Buffer 14 Mask Bits</p> <p>Each mask bit masks the corresponding Mailbox 14 filter field in the same way that RXMGMASK masks other Mailboxes' filters. See the description of the CAN_RXMGMASK register.</p> <p>0 The corresponding bit in the filter is "don't care."<br/>1 The corresponding bit in the filter is checked.</p> |

## 37.4.7 Rx 15 Mask register (CAN\_RX15MASK)

This register is located in RAM.

RX15MASK is provided for legacy application support. When the MCR[IRMQ] bit is asserted, RX15MASK has no effect.

RX15MASK is used to mask the filter fields of Message Buffer 15.

This register can be programmed only while the module is in Freeze mode because it is blocked by hardware in other modes.

Address: 0h base + 18h offset = 18h

|       |             |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0           | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | RX15M[31:0] |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |             |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 1           | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  |    |

### CAN\_RX15MASK field descriptions

| Field               | Description                                                                                                                                                                                                                                                                                                                            |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>RX15M[31:0] | <p>Rx Buffer 15 Mask Bits</p> <p>Each mask bit masks the corresponding Mailbox 15 filter field in the same way that RXMGMASK masks other Mailboxes' filters. See the description of the CAN_RXMGMASK register.</p> <p>0 The corresponding bit in the filter is "don't care."<br/>1 The corresponding bit in the filter is checked.</p> |

### 37.4.8 Error Counter (CAN\_ECR)

This register has two 8-bit fields reflecting the value of two FlexCAN error counters: Transmit Error Counter (TXERRCNT field) and Receive Error Counter (RXERRCNT field). The rules for increasing and decreasing these counters are described in the CAN protocol and are completely implemented in the FlexCAN module. Both counters are read-only except in Freeze mode, where they can be written by the CPU.

FlexCAN responds to any bus state as described in the protocol, for example, transmit Error Active or Error Passive flag, delay its transmission start time (Error Passive) and avoid any influence on the bus when in Bus Off state.

The following are the basic rules for FlexCAN bus state transitions:

- If the value of TXERRCNT or RXERRCNT increases to be greater than or equal to 128, the FLTCONF field in the Error and Status Register is updated to reflect 'Error Passive' state.
- If the FlexCAN state is 'Error Passive', and either TXERRCNT or RXERRCNT decrements to a value less than or equal to 127 while the other already satisfies this condition, the FLTCONF field in the Error and Status Register is updated to reflect 'Error Active' state.
- If the value of TXERRCNT increases to be greater than 255, the FLTCONF field in the Error and Status Register is updated to reflect 'Bus Off' state, and an interrupt may be issued. The value of TXERRCNT is then reset to zero.
- If FlexCAN is in 'Bus Off' state, then TXERRCNT is cascaded together with another internal counter to count the 128th occurrences of 11 consecutive recessive bits on the bus. Hence, TXERRCNT is reset to zero and counts in a manner where the internal counter counts 11 such bits and then wraps around while incrementing the TXERRCNT. When TXERRCNT reaches the value of 128, the FLTCONF field in the Error and Status Register is updated to be 'Error Active' and both error counters are reset to zero. At any instance of dominant bit following a stream of less than 11 consecutive recessive bits, the internal counter resets itself to zero without affecting the TXERRCNT value.
- If during system start-up, only one node is operating, then its TXERRCNT increases in each message it is trying to transmit, as a result of acknowledge errors (indicated by the ACKERR bit in the Error and Status Register). After the transition to 'Error Passive' state, the TXERRCNT does not increment anymore by acknowledge errors. Therefore the device never goes to the 'Bus Off' state.
- If the RXERRCNT increases to a value greater than 127, it is not incremented further, even if more errors are detected while being a receiver. At the next successful message reception, the counter is set to a value between 119 and 127 to resume to 'Error Active' state.

Address: 0h base + 1Ch offset = 1Ch

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |          |    |    |    |    |    |    |    |          |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----------|----|----|----|----|----|----|----|----------|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16       | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24       | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | RXERRCNT |    |    |    |    |    |    |    | TXERRCNT |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |          |    |    |    |    |    |    |    |          |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

### CAN\_ECR field descriptions

| Field             | Description                                                                             |
|-------------------|-----------------------------------------------------------------------------------------|
| 0–15<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 16–23<br>RXERRCNT | Receive Error Counter                                                                   |
| 24–31<br>TXERRCNT | Transmit Error Counter                                                                  |

## 37.4.9 Error and Status 1 register (CAN\_ESR1)

This register reflects various error conditions and some general status of the device. It is also the source of interrupts to the CPU.

A CPU read operation clears the following fields to 0, so these fields report error conditions that occurred after the last time the CPU read this register: BIT1ERR, BIT0ERR, ACKERR, CRCERR, FRMERR, and STFERR. TXWRN, RXWRN, IDLE, TX, FLTCONF, and RX provide status information.

The following table shows the FlexCAN state variables and their meanings. Combinations not shown in the table are reserved.

| SYNCH | IDLE | TX | RX | FlexCAN state               |
|-------|------|----|----|-----------------------------|
| 0     | 0    | 0  | 0  | Not synchronized to CAN bus |
| 1     | 1    | x  | x  | Idle                        |
| 1     | 0    | 1  | 0  | Transmitting                |
| 1     | 0    | 0  | 1  | Receiving                   |

## Memory map/register definition

Address: 0h base + 20h offset = 20h

|       |   |   |   |   |   |   |   |   |   |   |    |    |       |         |         |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|-------|---------|---------|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12    | 13      | 14      | 15 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    | SYNCH | TWRNINT | RWRNINT |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |       | w1c     | w1c     |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0     | 0       | 0       | 0  |

|       |         |         |        |        |        |        |       |       |      |    |         |    |         |        |    |    |
|-------|---------|---------|--------|--------|--------|--------|-------|-------|------|----|---------|----|---------|--------|----|----|
| Bit   | 16      | 17      | 18     | 19     | 20     | 21     | 22    | 23    | 24   | 25 | 26      | 27 | 28      | 29     | 30 | 31 |
| R     | BIT1ERR | BIT0ERR | ACKERR | CRCERR | FRMERR | STFERR | TXWRN | RXWRN | IDLE | TX | FLTCONF | RX | BOFFINT | ERRINT | 0  |    |
| W     |         |         |        |        |        |        |       |       |      |    |         |    | w1c     | w1c    |    |    |
| Reset | 0       | 0       | 0      | 0      | 0      | 0      | 0     | 0     | 0    | 0  | 0       | 0  | 0       | 0      | 0  | 0  |

### CAN\_ESR1 field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–12<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                        |
| 13<br>SYNCH      | CAN Synchronization Status<br><br>This read-only flag indicates whether the FlexCAN is synchronized to the CAN bus and able to participate in the communication process. It is set and cleared by the FlexCAN. See the table in the overall CAN_ESR1 register description.<br><br>0 FlexCAN is not synchronized to the CAN bus.<br>1 FlexCAN is synchronized to the CAN bus.                                                                                   |
| 14<br>TWRNINT    | Tx Warning Interrupt Flag<br><br>If the WRNEN bit in MCR is asserted, the TWRNINT bit is set when the TXWRN flag transitions from 0 to 1, meaning that the Tx error counter reached 96. If the corresponding mask bit in the Control Register (TWRNMSK) is set, an interrupt is generated to the CPU. This bit is cleared by writing it to 1. When WRNEN is negated, this flag is masked. CPU must clear this flag before disabling the bit. Otherwise it will |

Table continues on the next page...



## CAN\_ESR1 field descriptions (continued)

| Field         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               | <p>be set when the WRNEN is set again. Writing 0 has no effect. This flag is not generated during Bus Off state. This bit is not updated during Freeze mode.</p> <p>0 No such occurrence.<br/>1 The Tx error counter transitioned from less than 96 to greater than or equal to 96.</p>                                                                                                                                                                                                                                                                                                                                                                                                                |
| 15<br>RWRNINT | <p>Rx Warning Interrupt Flag</p> <p>If the WRNEN bit in MCR is asserted, the RWRNINT bit is set when the RXWRN flag transitions from 0 to 1, meaning that the Rx error counters reached 96. If the corresponding mask bit in the Control Register (RWRNMSK) is set, an interrupt is generated to the CPU. This bit is cleared by writing it to 1. When WRNEN is negated, this flag is masked. CPU must clear this flag before disabling the bit. Otherwise it will be set when the WRNEN is set again. Writing 0 has no effect. This bit is not updated during Freeze mode.</p> <p>0 No such occurrence.<br/>1 The Rx error counter transitioned from less than 96 to greater than or equal to 96.</p> |
| 16<br>BIT1ERR | <p>Bit1 Error</p> <p>This bit indicates when an inconsistency occurs between the transmitted and the received bit in a message.</p> <p><b>NOTE:</b> This bit is not set by a transmitter in case of arbitration field or ACK slot, or in case of a node sending a passive error flag that detects dominant bits.</p> <p>0 No such occurrence.<br/>1 At least one bit sent as recessive is received as dominant.</p>                                                                                                                                                                                                                                                                                    |
| 17<br>BIT0ERR | <p>Bit0 Error</p> <p>This bit indicates when an inconsistency occurs between the transmitted and the received bit in a message.</p> <p>0 No such occurrence.<br/>1 At least one bit sent as dominant is received as recessive.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 18<br>ACKERR  | <p>Acknowledge Error</p> <p>This bit indicates that an Acknowledge Error has been detected by the transmitter node, that is, a dominant bit has not been detected during the ACK SLOT.</p> <p>0 No such occurrence.<br/>1 An ACK error occurred since last read of this register.</p>                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 19<br>CRCERR  | <p>Cyclic Redundancy Check Error</p> <p>This bit indicates that a CRC Error has been detected by the receiver node, that is, the calculated CRC is different from the received.</p> <p>0 No such occurrence.<br/>1 A CRC error occurred since last read of this register.</p>                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 20<br>FRMERR  | <p>Form Error</p> <p>This bit indicates that a Form Error has been detected by the receiver node, that is, a fixed-form bit field contains at least one illegal bit.</p> <p>0 No such occurrence.<br/>1 A Form Error occurred since last read of this register.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                    |

Table continues on the next page...

**CAN\_ESR1 field descriptions (continued)**

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 21<br>STFERR     | <p>Stuffing Error</p> <p>This bit indicates that a Stuffing Error has been detected.</p> <p>0 No such occurrence.<br/>1 A Stuffing Error occurred since last read of this register.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 22<br>TXWRN      | <p>TX Error Warning</p> <p>This bit indicates when repetitive errors are occurring during message transmission. This bit is not updated during Freeze mode.</p> <p>0 No such occurrence.<br/>1 TXERRCNT is greater than or equal to 96.</p>                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 23<br>RXWRN      | <p>Rx Error Warning</p> <p>This bit indicates when repetitive errors are occurring during message reception. This bit is not updated during Freeze mode.</p> <p>0 No such occurrence.<br/>1 RXERRCNT is greater than or equal to 96.</p>                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 24<br>IDLE       | <p>This bit indicates when CAN bus is in IDLE state. See the table in the overall CAN_ESR1 register description.</p> <p>0 No such occurrence.<br/>1 CAN bus is now IDLE.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 25<br>TX         | <p>FlexCAN In Transmission</p> <p>This bit indicates if FlexCAN is transmitting a message. See the table in the overall CAN_ESR1 register description.</p> <p>0 FlexCAN is not transmitting a message.<br/>1 FlexCAN is transmitting a message.</p>                                                                                                                                                                                                                                                                                                                                                                                                                |
| 26–27<br>FLTCONF | <p>Fault Confinement State</p> <p>This 2-bit field indicates the Confinement State of the FlexCAN module.</p> <p>If the LOM bit in the Control Register is asserted, after some delay that depends on the CAN bit timing the FLTCONF field will indicate “Error Passive”. The very same delay affects the way how FLTCONF reflects an update to ECR register by the CPU. It may be necessary up to one CAN bit time to get them coherent again.</p> <p>Because the Control Register is not affected by soft reset, the FLTCONF field will not be affected by soft reset if the LOM bit is asserted.</p> <p>00 Error Active<br/>01 Error Passive<br/>1x Bus Off</p> |
| 28<br>RX         | <p>FlexCAN In Reception</p> <p>This bit indicates if FlexCAN is receiving a message. See the table in the overall CAN_ESR1 register description.</p> <p>0 FlexCAN is not receiving a message.<br/>1 FlexCAN is receiving a message.</p>                                                                                                                                                                                                                                                                                                                                                                                                                            |

*Table continues on the next page...*

## CAN\_ESR1 field descriptions (continued)

| Field          | Description                                                                                                                                                                                                                                                                                                                                                                  |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 29<br>BOFFINT  | <p>Bus Off Interrupt</p> <p>This bit is set when FlexCAN enters 'Bus Off' state. If the corresponding mask bit in the Control Register (BOFFMSK) is set, an interrupt is generated to the CPU. This bit is cleared by writing it to 1. Writing 0 has no effect.</p> <p>0 No such occurrence.<br/>1 FlexCAN module entered Bus Off state.</p>                                 |
| 30<br>ERRINT   | <p>Error Interrupt</p> <p>This bit indicates that at least one of the Error Bits (bits 15-10) is set. If the corresponding mask bit CTRL1[ERRMSK] is set, an interrupt is generated to the CPU. This bit is cleared by writing it to 1. Writing 0 has no effect.</p> <p>0 No such occurrence.<br/>1 Indicates setting of any Error Bit in the Error and Status Register.</p> |
| 31<br>Reserved | <p>This field is reserved.<br/>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                              |

## 37.4.10 Interrupt Masks 2 register (CAN\_IMASK2)

This register allows any number of a range of the 32 Message Buffer Interrupts to be enabled or disabled for MB63 to MB32. It contains one interrupt mask bit per buffer, enabling the CPU to determine which buffer generates an interrupt after a successful transmission or reception, that is, when the corresponding IFLAG2 bit is set.

Address: 0h base + 24h offset = 24h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

## CAN\_IMASK2 field descriptions

| Field         | Description                                                                                                                                                                                                                                                                                                                                                                                                      |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>BUFHM | <p>Buffer MB<sub>i</sub> Mask</p> <p>Each bit enables or disables the corresponding FlexCAN Message Buffer Interrupt for MB63 to MB32.</p> <p><b>NOTE:</b> Setting or clearing a bit in the IMASK2 Register can assert or negate an interrupt request, if the corresponding IFLAG2 bit is set.</p> <p>0 The corresponding buffer Interrupt is disabled.<br/>1 The corresponding buffer Interrupt is enabled.</p> |

### 37.4.11 Interrupt Masks 1 register (CAN\_IMASK1)

This register allows any number of a range of the 32 Message Buffer Interrupts to be enabled or disabled for MB31 to MB0. It contains one interrupt mask bit per buffer, enabling the CPU to determine which buffer generates an interrupt after a successful transmission or reception, that is, when the corresponding IFLAG1 bit is set.

Address: 0h base + 28h offset = 28h

|       |       |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0     | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | BUFLM |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |       |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0     | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

#### CAN\_IMASK1 field descriptions

| Field         | Description                                                                                                                                                                                                                                                                                                                                                                                                        |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>BUFLM | <p>Buffer MB<sub>i</sub> Mask</p> <p>Each bit enables or disables the corresponding FlexCAN Message Buffer Interrupt for MB31 to MB0.</p> <p><b>NOTE:</b> Setting or clearing a bit in the IMASK1 Register can assert or negate an interrupt request, if the corresponding IFLAG1 bit is set.</p> <p>0 The corresponding buffer Interrupt is disabled.</p> <p>1 The corresponding buffer Interrupt is enabled.</p> |

### 37.4.12 Interrupt Flags 2 register (CAN\_IFLAG2)

This register defines the flags for the 32 Message Buffer interrupts for MB63 to MB32. It contains one interrupt flag bit per buffer. Each successful transmission or reception sets the corresponding IFLAG2 bit. If the corresponding IMASK2 bit is set, an interrupt will be generated. The interrupt flag must be cleared by writing 1 to it. Writing 0 has no effect.

Before updating MCR[MAXMB] field, CPU must service the IFLAG2 bits whose MB value is greater than the MCR[MAXMB] to be updated; otherwise, they will remain set and be inconsistent with the number of MBs available.

Address: 0h base + 2Ch offset = 2Ch

|       |       |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0     | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | BUFHI |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     | w1c   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0     | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**CAN\_IFLAG2 field descriptions**

| Field         | Description                                                                                                                                                                                                                                                                                                                        |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>BUFHI | <p>Buffer MB<sub>i</sub> Interrupt</p> <p>Each bit flags the corresponding FlexCAN Message Buffer interrupt for MB63 to MB32.</p> <p>0 The corresponding buffer has no occurrence of successfully completed transmission or reception.</p> <p>1 The corresponding buffer has successfully completed transmission or reception.</p> |

**37.4.13 Interrupt Flags 1 register (CAN\_IFLAG1)**

This register defines the flags for the 32 Message Buffer interrupts for MB31 to MB0. It contains one interrupt flag bit per buffer. Each successful transmission or reception sets the corresponding IFLAG1 bit. If the corresponding IMASK1 bit is set, an interrupt will be generated. The interrupt flag must be cleared by writing 1 to it. Writing 0 has no effect.

The BUF7I to BUF5I flags are also used to represent FIFO interrupts when the Rx FIFO is enabled. When the bit MCR[RFEN] is set, the function of the 8 least significant interrupt flags BUF[7:0]I changes: BUF7I, BUF6I and BUF5I indicate operating conditions of the FIFO, and the BUF4TO0I field is reserved.

Before enabling the RFEN, the CPU must service the IFLAG bits asserted in the Rx FIFO region; see Section "Rx FIFO". Otherwise, these IFLAG bits will mistakenly show the related MBs now belonging to FIFO as having contents to be serviced. When the RFEN bit is negated, the FIFO flags must be cleared. The same care must be taken when an RFFN value is selected extending Rx FIFO filters beyond MB7. For example, when RFFN is 0x8, the MB0-23 range is occupied by Rx FIFO filters and related IFLAG bits must be cleared.

Before updating MCR[MAXMB] field, CPU must service the IFLAG1 bits whose MB value is greater than the MCR[MAXMB] to be updated; otherwise, they will remain set and be inconsistent with the number of MBs available.

Address: 0h base + 30h offset = 30h

|       |           |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|-----------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0         | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | BUF31TO8I |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     | w1c       |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0         | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

## Memory map/register definition

| Bit   | 16        | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24    | 25    | 26    | 27       | 28 | 29 | 30 | 31    |
|-------|-----------|----|----|----|----|----|----|----|-------|-------|-------|----------|----|----|----|-------|
| R     | BUF31TO8I |    |    |    |    |    |    |    | BUF7I | BUF6I | BUF5I | BUF4TO1I |    |    |    | BUF0I |
| W     | w1c       |    |    |    |    |    |    |    | w1c   | w1c   | w1c   | w1c      |    |    |    | w1c   |
| Reset | 0         | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0     | 0     | 0     | 0        | 0  | 0  | 0  | 0     |

### CAN\_IFLAG1 field descriptions

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–23<br>BUF31TO8I | <p>Buffer MB<sub>i</sub> Interrupt</p> <p>Each bit flags the corresponding FlexCAN Message Buffer interrupt for MB31 to MB8.</p> <p>0 The corresponding buffer has no occurrence of successfully completed transmission or reception.<br/>1 The corresponding buffer has successfully completed transmission or reception.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 24<br>BUF7I       | <p>Buffer MB7 Interrupt Or "Rx FIFO Overflow"</p> <p>When the RFEN bit in the MCR is cleared (Rx FIFO disabled), this bit flags the interrupt for MB7.</p> <p><b>NOTE:</b> This flag is cleared by the FlexCAN whenever the bit MCR[RFEN] is changed by CPU writes.</p> <p>The BUF7I flag represents "Rx FIFO Overflow" when MCR[RFEN] is set. In this case, the flag indicates that a message was lost because the Rx FIFO is full. Note that the flag will not be asserted when the Rx FIFO is full and the message was captured by a Mailbox.</p> <p>0 No occurrence of MB7 completing transmission/reception when MCR[RFEN]=0, or of Rx FIFO overflow when MCR[RFEN]=1<br/>1 MB7 completed transmission/reception when MCR[RFEN]=0, or Rx FIFO overflow when MCR[RFEN]=1</p>                                                                                                                                                                                                                   |
| 25<br>BUF6I       | <p>Buffer MB6 Interrupt Or "Rx FIFO Warning"</p> <p>When the RFEN bit in the MCR is cleared (Rx FIFO disabled), this bit flags the interrupt for MB6.</p> <p><b>NOTE:</b> This flag is cleared by the FlexCAN whenever the bit MCR[RFEN] is changed by CPU writes.</p> <p>The BUF6I flag represents "Rx FIFO Warning" when MCR[RFEN] is set. In this case, the flag indicates when the number of unread messages within the Rx FIFO is increased to 5 from 4 due to the reception of a new one, meaning that the Rx FIFO is almost full. Note that if the flag is cleared while the number of unread messages is greater than 4, it does not assert again until the number of unread messages within the Rx FIFO is decreased to be equal to or less than 4.</p> <p>0 No occurrence of MB6 completing transmission/reception when MCR[RFEN]=0, or of Rx FIFO almost full when MCR[RFEN]=1<br/>1 MB6 completed transmission/reception when MCR[RFEN]=0, or Rx FIFO almost full when MCR[RFEN]=1</p> |
| 26<br>BUF5I       | <p>Buffer MB5 Interrupt Or "Frames available in Rx FIFO"</p> <p>When the RFEN bit in the MCR is cleared (Rx FIFO disabled), this bit flags the interrupt for MB5.</p> <p><b>NOTE:</b> This flag is cleared by the FlexCAN whenever the bit MCR[RFEN] is changed by CPU writes.</p> <p>The BUF5I flag represents "Frames available in Rx FIFO" when MCR[RFEN] is set. In this case, the flag indicates that at least one frame is available to be read from the Rx FIFO.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |

Table continues on the next page...

## CAN\_IFLAG1 field descriptions (continued)

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | <p>0 No occurrence of MB5 completing transmission/reception when MCR[RFEN]=0, or of frame(s) available in the FIFO, when MCR[RFEN]=1</p> <p>1 MB5 completed transmission/reception when MCR[RFEN]=0, or frame(s) available in the Rx FIFO when MCR[RFEN]=1</p>                                                                                                                                                                                                                                                                                                                   |
| 27–30<br>BUF4TO1I | <p>Buffer MB<sub>i</sub> Interrupt Or "reserved"</p> <p>When the RFEN bit in the MCR is cleared (Rx FIFO disabled), these bits flag the interrupts for MB4 to MB1.</p> <p><b>NOTE:</b> These flags are cleared by the FlexCAN whenever the bit MCR[RFEN] is changed by CPU writes. The BUF4TO1I flags are reserved when MCR[RFEN] is set.</p> <p>0 The corresponding buffer has no occurrence of successfully completed transmission or reception when MCR[RFEN]=0.</p> <p>1 The corresponding buffer has successfully completed transmission or reception when MCR[RFEN]=0.</p> |
| 31<br>BUF0I       | <p>Buffer MB0 Interrupt Or "reserved"</p> <p>When the RFEN bit in the MCR is cleared (Rx FIFO disabled), this bit flags the interrupt for MB0.</p> <p><b>NOTE:</b> This flag is cleared by the FlexCAN whenever the bit MCR[RFEN] is changed by CPU writes. The BUF0I flag is reserved when MCR[RFEN] is set.</p> <p>0 The corresponding buffer has no occurrence of successfully completed transmission or reception when MCR[RFEN]=0.</p> <p>1 The corresponding buffer has successfully completed transmission or reception when MCR[RFEN]=0.</p>                             |

## 37.4.14 Control 2 register (CAN\_CTRL2)

This register contains control bits for CAN errors, FIFO features, and mode selection.

Address: 0h base + 34h offset = 34h

|       |    |    |       |        |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----|----|-------|--------|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0  | 1  | 2     | 3      | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0  |    | ECRWE | WRMFRZ |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |    |    |       |        |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0  | 0  | 0     | 0      | 0  | 0  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| Bit   | 16 | 17 | 18    | 19     | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |    |    |       |        |    |    |    |    | 0  |    |    |    |    |    |    |    |
| W     |    |    |       |        |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0  | 0  | 0     | 0      | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

## CAN\_CTRL2 field descriptions

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                         |                                            |                                                                  |                                                                               |                                                                  |                                                                               |     |   |        |         |              |      |     |    |        |          |              |                |     |    |         |          |               |                |     |    |         |          |               |                |     |    |         |          |               |                |     |    |         |          |               |                |     |    |         |          |               |                |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------|--------------------------------------------|------------------------------------------------------------------|-------------------------------------------------------------------------------|------------------------------------------------------------------|-------------------------------------------------------------------------------|-----|---|--------|---------|--------------|------|-----|----|--------|----------|--------------|----------------|-----|----|---------|----------|---------------|----------------|-----|----|---------|----------|---------------|----------------|-----|----|---------|----------|---------------|----------------|-----|----|---------|----------|---------------|----------------|-----|----|---------|----------|---------------|----------------|
| 0–1<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                                                         |                                            |                                                                  |                                                                               |                                                                  |                                                                               |     |   |        |         |              |      |     |    |        |          |              |                |     |    |         |          |               |                |     |    |         |          |               |                |     |    |         |          |               |                |     |    |         |          |               |                |     |    |         |          |               |                |
| 2<br>ECRWRE     | Error-correction Configuration Register Write Enable<br><br>Enable that the MECR register is updated. This bit is automatically set to 0 if the protocol described in the "Detection and Correction of Memory Errors" section is not followed.<br><br>0   Disable update.<br>1   Enable update.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                         |                                            |                                                                  |                                                                               |                                                                  |                                                                               |     |   |        |         |              |      |     |    |        |          |              |                |     |    |         |          |               |                |     |    |         |          |               |                |     |    |         |          |               |                |     |    |         |          |               |                |     |    |         |          |               |                |
| 3<br>WRMFRZ     | Write-Access To Memory In Freeze Mode<br><br>Enable unrestricted write access to FlexCAN memory in Freeze mode. This bit can only be written in Freeze mode and has no effect out of Freeze mode.<br><br>0   Maintain the write access restrictions.<br>1   Enable unrestricted write access to FlexCAN memory.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                         |                                            |                                                                  |                                                                               |                                                                  |                                                                               |     |   |        |         |              |      |     |    |        |          |              |                |     |    |         |          |               |                |     |    |         |          |               |                |     |    |         |          |               |                |     |    |         |          |               |                |     |    |         |          |               |                |
| 4–7<br>RFFN     | <p>Number Of Rx FIFO Filters</p> <p>This 4-bit field defines the number of Rx FIFO filters, as shown in the following table. The maximum selectable number of filters is determined by the chip. This field can only be written in Freeze mode as it is blocked by hardware in other modes. This field must not be programmed with values that make the number of Message Buffers occupied by Rx FIFO and ID Filter exceed the number of Mailboxes present, defined by MCR[MAXMB].</p> <p><b>NOTE:</b> Each group of eight filters occupies a memory space equivalent to two Message Buffers which means that the more filters are implemented the less Mailboxes will be available.</p> <p>Considering that the Rx FIFO occupies the memory space originally reserved for MB0-5, RFFN should be programmed with a value corresponding to a number of filters not greater than the number of available memory words which can be calculated as follows:</p> <p><math>(\text{SETUP\_MB} - 6) \times 4</math></p> <p>where SETUP_MB is the least between NUMBER_OF_MB and MAXMB.</p> <p>The number of remaining Mailboxes available will be:</p> <p><math>(\text{SETUP\_MB} - 8) - (\text{RFFN} \times 2)</math></p> <p>If the Number of Rx FIFO Filters programmed through RFFN exceeds the SETUP_MB value (memory space available) the exceeding ones will not be functional.</p> <table><tr><th>RFFN[3:0]</th><th>Number of Rx FIFO filters</th><th>Message Buffers occupied by Rx FIFO and ID Filter Table</th><th>Remaining Available Mailboxes<sup>1</sup></th><th>Rx FIFO ID Filter Table Elements Affected by Rx Individual Masks</th><th>Rx FIFO ID Filter Table Elements Affected by Rx FIFO Global Mask<sup>2</sup></th></tr><tr><td>0x0</td><td>8</td><td>MB 0-7</td><td>MB 8-63</td><td>Elements 0-7</td><td>none</td></tr><tr><td>0x1</td><td>16</td><td>MB 0-9</td><td>MB 10-63</td><td>Elements 0-9</td><td>Elements 10-15</td></tr><tr><td>0x2</td><td>24</td><td>MB 0-11</td><td>MB 12-63</td><td>Elements 0-11</td><td>Elements 12-23</td></tr><tr><td>0x3</td><td>32</td><td>MB 0-13</td><td>MB 14-63</td><td>Elements 0-13</td><td>Elements 14-31</td></tr><tr><td>0x4</td><td>40</td><td>MB 0-15</td><td>MB 16-63</td><td>Elements 0-15</td><td>Elements 16-39</td></tr><tr><td>0x5</td><td>48</td><td>MB 0-17</td><td>MB 18-63</td><td>Elements 0-17</td><td>Elements 18-47</td></tr><tr><td>0x6</td><td>56</td><td>MB 0-19</td><td>MB 20-63</td><td>Elements 0-19</td><td>Elements 20-55</td></tr></table> | RFFN[3:0]                                               | Number of Rx FIFO filters                  | Message Buffers occupied by Rx FIFO and ID Filter Table          | Remaining Available Mailboxes <sup>1</sup>                                    | Rx FIFO ID Filter Table Elements Affected by Rx Individual Masks | Rx FIFO ID Filter Table Elements Affected by Rx FIFO Global Mask <sup>2</sup> | 0x0 | 8 | MB 0-7 | MB 8-63 | Elements 0-7 | none | 0x1 | 16 | MB 0-9 | MB 10-63 | Elements 0-9 | Elements 10-15 | 0x2 | 24 | MB 0-11 | MB 12-63 | Elements 0-11 | Elements 12-23 | 0x3 | 32 | MB 0-13 | MB 14-63 | Elements 0-13 | Elements 14-31 | 0x4 | 40 | MB 0-15 | MB 16-63 | Elements 0-15 | Elements 16-39 | 0x5 | 48 | MB 0-17 | MB 18-63 | Elements 0-17 | Elements 18-47 | 0x6 | 56 | MB 0-19 | MB 20-63 | Elements 0-19 | Elements 20-55 |
| RFFN[3:0]       | Number of Rx FIFO filters                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Message Buffers occupied by Rx FIFO and ID Filter Table | Remaining Available Mailboxes <sup>1</sup> | Rx FIFO ID Filter Table Elements Affected by Rx Individual Masks | Rx FIFO ID Filter Table Elements Affected by Rx FIFO Global Mask <sup>2</sup> |                                                                  |                                                                               |     |   |        |         |              |      |     |    |        |          |              |                |     |    |         |          |               |                |     |    |         |          |               |                |     |    |         |          |               |                |     |    |         |          |               |                |     |    |         |          |               |                |
| 0x0             | 8                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | MB 0-7                                                  | MB 8-63                                    | Elements 0-7                                                     | none                                                                          |                                                                  |                                                                               |     |   |        |         |              |      |     |    |        |          |              |                |     |    |         |          |               |                |     |    |         |          |               |                |     |    |         |          |               |                |     |    |         |          |               |                |     |    |         |          |               |                |
| 0x1             | 16                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | MB 0-9                                                  | MB 10-63                                   | Elements 0-9                                                     | Elements 10-15                                                                |                                                                  |                                                                               |     |   |        |         |              |      |     |    |        |          |              |                |     |    |         |          |               |                |     |    |         |          |               |                |     |    |         |          |               |                |     |    |         |          |               |                |     |    |         |          |               |                |
| 0x2             | 24                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | MB 0-11                                                 | MB 12-63                                   | Elements 0-11                                                    | Elements 12-23                                                                |                                                                  |                                                                               |     |   |        |         |              |      |     |    |        |          |              |                |     |    |         |          |               |                |     |    |         |          |               |                |     |    |         |          |               |                |     |    |         |          |               |                |     |    |         |          |               |                |
| 0x3             | 32                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | MB 0-13                                                 | MB 14-63                                   | Elements 0-13                                                    | Elements 14-31                                                                |                                                                  |                                                                               |     |   |        |         |              |      |     |    |        |          |              |                |     |    |         |          |               |                |     |    |         |          |               |                |     |    |         |          |               |                |     |    |         |          |               |                |     |    |         |          |               |                |
| 0x4             | 40                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | MB 0-15                                                 | MB 16-63                                   | Elements 0-15                                                    | Elements 16-39                                                                |                                                                  |                                                                               |     |   |        |         |              |      |     |    |        |          |              |                |     |    |         |          |               |                |     |    |         |          |               |                |     |    |         |          |               |                |     |    |         |          |               |                |     |    |         |          |               |                |
| 0x5             | 48                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | MB 0-17                                                 | MB 18-63                                   | Elements 0-17                                                    | Elements 18-47                                                                |                                                                  |                                                                               |     |   |        |         |              |      |     |    |        |          |              |                |     |    |         |          |               |                |     |    |         |          |               |                |     |    |         |          |               |                |     |    |         |          |               |                |     |    |         |          |               |                |
| 0x6             | 56                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | MB 0-19                                                 | MB 20-63                                   | Elements 0-19                                                    | Elements 20-55                                                                |                                                                  |                                                                               |     |   |        |         |              |      |     |    |        |          |              |                |     |    |         |          |               |                |     |    |         |          |               |                |     |    |         |          |               |                |     |    |         |          |               |                |     |    |         |          |               |                |

Table continues on the next page...



## CAN\_CTRL2 field descriptions (continued)

| Field        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                           |                                                         |                                            |                                                                  |                                                                               |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------|---------------------------------------------------------|--------------------------------------------|------------------------------------------------------------------|-------------------------------------------------------------------------------|
|              | RFFN[3:0]                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Number of Rx FIFO filters | Message Buffers occupied by Rx FIFO and ID Filter Table | Remaining Available Mailboxes <sup>1</sup> | Rx FIFO ID Filter Table Elements Affected by Rx Individual Masks | Rx FIFO ID Filter Table Elements Affected by Rx FIFO Global Mask <sup>2</sup> |
|              | 0x7                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 64                        | MB 0-21                                                 | MB 22-63                                   | Elements 0-21                                                    | Elements 22-63                                                                |
|              | 0x8                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 72                        | MB 0-23                                                 | MB 24-63                                   | Elements 0-23                                                    | Elements 24-71                                                                |
|              | 0x9                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 80                        | MB 0-25                                                 | MB 26-63                                   | Elements 0-25                                                    | Elements 26-79                                                                |
|              | 0xA                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 88                        | MB 0-27                                                 | MB 28-63                                   | Elements 0-27                                                    | Elements 28-87                                                                |
|              | 0xB                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 96                        | MB 0-29                                                 | MB 30-63                                   | Elements 0-29                                                    | Elements 30-95                                                                |
|              | 0xC                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 104                       | MB 0-31                                                 | MB 32-63                                   | Elements 0-31                                                    | Elements 32-103                                                               |
|              | 0xD                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 112                       | MB 0-33                                                 | MB 34-63                                   | Elements 0-31                                                    | Elements 32-111                                                               |
|              | 0xE                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 120                       | MB 0-35                                                 | MB 36-63                                   | Elements 0-31                                                    | Elements 32-119                                                               |
|              | 0xF                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 128                       | MB 0-37                                                 | MB 38-63                                   | Elements 0-31                                                    | Elements 32-127                                                               |
|              | <p>1. The number of the last remaining available mailboxes is defined by the least value between the parameter NUMBER_OF_MB minus 1 and the MCR[MAXMB] field.</p> <p>2. If Rx Individual Mask Registers are not enabled then all Rx FIFO filters are affected by the Rx FIFO Global Mask.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                           |                                                         |                                            |                                                                  |                                                                               |
| 8–12<br>TASD | <p><b>Tx Arbitration Start Delay</b><br/>This 5-bit field indicates how many CAN bits the Tx arbitration process start point can be delayed from the first bit of CRC field on CAN bus. This field can be written only in Freeze mode because it is blocked by hardware in other modes.</p> <p>This field is useful to optimize the transmit performance based on factors such as: peripheral/serial clock ratio, CAN bit timing and number of MBs. The duration of an arbitration process, in terms of CAN bits, is directly proportional to the number of available MBs and CAN baud rate and inversely proportional to the peripheral clock frequency.</p> <p>The optimal arbitration timing is that in which the last MB is scanned right before the first bit of the Intermission field of a CAN frame. Therefore, if there are few MBs and the system/serial clock ratio is high and the CAN baud rate is low then the arbitration can be delayed and vice-versa.</p> <p>If TASD is 0 then the arbitration start is not delayed, thus the CPU has less time to configure a Tx MB for the next arbitration, but more time is reserved for arbitration. On the other hand, if TASD is 24 then the CPU can configure a Tx MB later and less time is reserved for arbitration.</p> <p>If too little time is reserved for arbitration the FlexCAN may be not able to find winner MBs in time to compete with other nodes for the CAN bus. If the arbitration ends too much time before the first bit of Intermission field then there is a chance that the CPU reconfigures some Tx MBs and the winner MB is not the best to be transmitted.</p> <p>The optimal configuration for TASD can be calculated as:</p> $TASD = 25 - \left\{ f \times [MAXMB + 3 - \frac{CANCLK}{(RFEN \times 8)} - (RFEN \times RFFN \times 2)] \times 2 \right\} / \left\{ f \times [1 + (PSEG1+1) + \frac{SYS}{(PSEG2+1)} + (PROPSEG+1)] \times (PRES DIV+1) \right\}$ <p>where:</p> |                           |                                                         |                                            |                                                                  |                                                                               |

Table continues on the next page...

**CAN\_CTRL2 field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | <ul style="list-style-type: none"> <li>• <math>f_{CANCLK}</math> is the Protocol Engine (PE) Clock (see section "Protocol Timing"), in Hz</li> <li>• <math>f_{SYS}</math> is the peripheral clock, in Hz</li> <li>• MAXMB is the value in CTRL1[MAXMB] field</li> <li>• RFEN is the value in CTRL1[RFEN] bit</li> <li>• RFFN is the value in CTRL2[RFFN] field</li> <li>• PSEG1 is the value in CTRL1[PSEG1] field</li> <li>• PSEG2 is the value in CTRL1[PSEG2] field</li> <li>• PROPSEG is the value in CTRL1[PROPSEG] field</li> <li>• PRESDIV is the value in CTRL1[PRES DIV] field</li> </ul> <p>See Section "Arbitration process" and Section "Protocol Timing" for more details.</p> |
| 13<br>MRP         | <p>Mailboxes Reception Priority</p> <p>If this bit is set the matching process starts from the Mailboxes and if no match occurs the matching continues on the Rx FIFO. This bit can be written only in Freeze mode because it is blocked by hardware in other modes.</p> <p>0 Matching starts from Rx FIFO and continues on Mailboxes.<br/>1 Matching starts from Mailboxes and continues on Rx FIFO.</p>                                                                                                                                                                                                                                                                                   |
| 14<br>RRS         | <p>Remote Request Storing</p> <p>If this bit is asserted Remote Request Frame is submitted to a matching process and stored in the corresponding Message Buffer in the same fashion of a Data Frame. No automatic Remote Response Frame will be generated.</p> <p>If this bit is negated the Remote Request Frame is submitted to a matching process and an automatic Remote Response Frame is generated if a Message Buffer with CODE=0b1010 is found with the same ID.</p> <p>This bit can be written only in Freeze mode because it is blocked by hardware in other modes.</p> <p>0 Remote Response Frame is generated.<br/>1 Remote Request Frame is stored.</p>                        |
| 15<br>EACEN       | <p>Entire Frame Arbitration Field Comparison Enable For Rx Mailboxes</p> <p>This bit controls the comparison of IDE and RTR bits within Rx Mailboxes filters with their corresponding bits in the incoming frame by the matching process. This bit does not affect matching for Rx FIFO. This bit can be written only in Freeze mode because it is blocked by hardware in other modes.</p> <p>0 Rx Mailbox filter's IDE bit is always compared and RTR is never compared despite mask bits.<br/>1 Enables the comparison of both Rx Mailbox filter's IDE and RTR bit with their corresponding bits within the incoming frame. Mask bits do apply.</p>                                       |
| 16–31<br>Reserved | <p>This field is reserved.<br/>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

1. The number of the last remaining available mailboxes is defined by the least value between the parameter NUMBER\_OF\_MB minus 1 and the MCR[MAXMB] field.
2. If Rx Individual Mask Registers are not enabled then all Rx FIFO filters are affected by the Rx FIFO Global Mask.

### 37.4.15 Error and Status 2 register (CAN\_ESR2)

This register reflects various interrupt flags and some general status.

Address: 0h base + 38h offset = 38h

|       |   |   |   |   |   |   |   |   |   |      |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|------|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9    | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0 |   |   |   |   |   |   |   |   | LPTM |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |      |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0    | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |     |     |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----|-----|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 16 | 17  | 18  | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0  | VPS | IMB | 0  |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |    |     |     |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0  | 0   | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**CAN\_ESR2 field descriptions**

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–8<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 9–15<br>LPTM    | Lowest Priority Tx Mailbox<br><br>If ESR2[VPS] is asserted, this field indicates the lowest number inactive Mailbox (see the IMB bit description). If there is no inactive Mailbox then the Mailbox indicated depends on CTRL1[LBUF] bit value. If CTRL1[LBUF] bit is negated then the Mailbox indicated is the one that has the greatest arbitration value (see the "Highest priority Mailbox first" section). If CTRL1[LBUF] bit is asserted then the Mailbox indicated is the highest number active Tx Mailbox. If a Tx Mailbox is being transmitted it is not considered in LPTM calculation. If ESR2[IMB] is not asserted and a frame is transmitted successfully, LPTM is updated with its Mailbox number.                                                                                                                                                                                                                                                                            |
| 16<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 17<br>VPS       | Valid Priority Status<br><br>This bit indicates whether IMB and LPTM contents are currently valid or not. VPS is asserted upon every complete Tx arbitration process unless the CPU writes to Control and Status word of a Mailbox that has already been scanned, that is, it is behind Tx Arbitration Pointer, during the Tx arbitration process. If there is no inactive Mailbox and only one Tx Mailbox that is being transmitted then VPS is not asserted. VPS is negated upon the start of every Tx arbitration process or upon a write to Control and Status word of any Mailbox.<br><br><b>NOTE:</b> ESR2[VPS] is not affected by any CPU write into Control Status (C/S) of a MB that is blocked by abort mechanism. When MCR[AEN] is asserted, the abort code write in C/S of a MB that is being transmitted (pending abort), or any write attempt into a Tx MB with IFLAG set is blocked.<br><br>0 Contents of IMB and LPTM are invalid.<br>1 Contents of IMB and LPTM are valid. |
| 18<br>IMB       | Inactive Mailbox<br><br>If ESR2[VPS] is asserted, this bit indicates whether there is any inactive Mailbox (CODE field is either 0b1000 or 0b0000). This bit is asserted in the following cases:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

*Table continues on the next page...*

## CAN\_ESR2 field descriptions (continued)

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | <ul style="list-style-type: none"> <li>During arbitration, if an LPTM is found and it is inactive.</li> <li>If IMB is not asserted and a frame is transmitted successfully.</li> </ul> <p>This bit is cleared in all start of arbitration (see Section "Arbitration process").</p> <p><b>NOTE:</b> LPTM mechanism have the following behavior: if an MB is successfully transmitted and ESR2[IMB]=0 (no inactive Mailbox), then ESR2[VPS] and ESR2[IMB] are asserted and the index related to the MB just transmitted is loaded into ESR2[LPTM].</p> <p>0 If ESR2[VPS] is asserted, the ESR2[LPTM] is not an inactive Mailbox.<br/> 1 If ESR2[VPS] is asserted, there is at least one inactive Mailbox. LPTM content is the number of the first one.</p> |
| 19–31<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |

## 37.4.16 CRC Register (CAN\_CRCCR)

This register provides information about the CRC of transmitted messages.

Address: 0h base + 44h offset = 44h

|       |   |   |   |   |   |   |   |   |       |   |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|-------|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8     | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0 |   |   |   |   |   |   |   | MBCRC |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |       |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0     | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 16 | 17    | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0  | TXCRC |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |    |       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0  | 0     | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

## CAN\_CRCCR field descriptions

| Field           | Description                                                                                                                                                                 |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–8<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                          |
| 9–15<br>MBCRC   | <p>CRC Mailbox</p> <p>This field indicates the number of the Mailbox corresponding to the value in TXCRC field.</p>                                                         |
| 16<br>Reserved  | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                          |
| 17–31<br>TXCRC  | <p>CRC Transmitted</p> <p>This field indicates the CRC value of the last message transmitted. This field is updated at the same time the Tx Interrupt Flag is asserted.</p> |

### 37.4.17 Rx FIFO Global Mask register (CAN\_RXFGMASK)

This register is located in RAM.

If Rx FIFO is enabled RXFGMASK is used to mask the Rx FIFO ID Filter Table elements that do not have a corresponding RXIMR according to CTRL2[RFFN] field setting.

This register can only be written in Freeze mode as it is blocked by hardware in other modes.

Address: 0h base + 48h offset = 48h

|       |           |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-----------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0         | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |           |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     | FGM[31:0] |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 1         | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  |    |

#### CAN\_RXFGMASK field descriptions

| Field                                               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                                                     |                                     |                          |                                                      |          |  |  |     |     |       |                    |                    |          |   |         |         |           |   |   |        |   |                     |                     |   |                          |                                                      |   |   |   |   |   |
|-----------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------|-------------------------------------|--------------------------|------------------------------------------------------|----------|--|--|-----|-----|-------|--------------------|--------------------|----------|---|---------|---------|-----------|---|---|--------|---|---------------------|---------------------|---|--------------------------|------------------------------------------------------|---|---|---|---|---|
| 0–31<br>FGM[31:0]                                   | <p>Rx FIFO Global Mask Bits</p> <p>These bits mask the ID Filter Table elements bits in a perfect alignment.</p> <p>The following table shows how the FGM bits correspond to each IDAF field.</p> <table><tr><th rowspan="2">Rx FIFO ID Filter Table Elements Format (MCR[IDAM])</th><th colspan="6">Identifier Acceptance Filter Fields</th></tr><tr><th>RTR</th><th>IDE</th><th>RXIDA</th><th>RXIDB <sup>1</sup></th><th>RXIDC <sup>2</sup></th><th>Reserved</th></tr><tr><td>A</td><td>FGM[31]</td><td>FGM[30]</td><td>FGM[29:1]</td><td>-</td><td>-</td><td>FGM[0]</td></tr><tr><td>B</td><td>FGM[31],<br/>FGM[15]</td><td>FGM[30],<br/>FGM[14]</td><td rowspan="2">-</td><td>FGM[29:16],<br/>FGM[13:0]</td><td rowspan="2">FGM[31:24],<br/>FGM[23:16],<br/>FGM[15:8],<br/>FGM[7:0]</td><td rowspan="2">-</td></tr><tr><td>C</td><td>-</td><td>-</td><td>-</td></tr></table> <p>1. If MCR[IDAM] field is equivalent to the format B only the fourteen most significant bits of the Identifier of the incoming frame are compared with the Rx FIFO filter.</p> <p>2. If MCR[IDAM] field is equivalent to the format C only the eight most significant bits of the Identifier of the incoming frame are compared with the Rx FIFO filter.</p> <p>0 The corresponding bit in the filter is "don't care."</p> <p>1 The corresponding bit in the filter is checked.</p> | Rx FIFO ID Filter Table Elements Format (MCR[IDAM]) | Identifier Acceptance Filter Fields |                          |                                                      |          |  |  | RTR | IDE | RXIDA | RXIDB <sup>1</sup> | RXIDC <sup>2</sup> | Reserved | A | FGM[31] | FGM[30] | FGM[29:1] | - | - | FGM[0] | B | FGM[31],<br>FGM[15] | FGM[30],<br>FGM[14] | - | FGM[29:16],<br>FGM[13:0] | FGM[31:24],<br>FGM[23:16],<br>FGM[15:8],<br>FGM[7:0] | - | C | - | - | - |
| Rx FIFO ID Filter Table Elements Format (MCR[IDAM]) | Identifier Acceptance Filter Fields                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                                                     |                                     |                          |                                                      |          |  |  |     |     |       |                    |                    |          |   |         |         |           |   |   |        |   |                     |                     |   |                          |                                                      |   |   |   |   |   |
|                                                     | RTR                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | IDE                                                 | RXIDA                               | RXIDB <sup>1</sup>       | RXIDC <sup>2</sup>                                   | Reserved |  |  |     |     |       |                    |                    |          |   |         |         |           |   |   |        |   |                     |                     |   |                          |                                                      |   |   |   |   |   |
| A                                                   | FGM[31]                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | FGM[30]                                             | FGM[29:1]                           | -                        | -                                                    | FGM[0]   |  |  |     |     |       |                    |                    |          |   |         |         |           |   |   |        |   |                     |                     |   |                          |                                                      |   |   |   |   |   |
| B                                                   | FGM[31],<br>FGM[15]                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | FGM[30],<br>FGM[14]                                 | -                                   | FGM[29:16],<br>FGM[13:0] | FGM[31:24],<br>FGM[23:16],<br>FGM[15:8],<br>FGM[7:0] | -        |  |  |     |     |       |                    |                    |          |   |         |         |           |   |   |        |   |                     |                     |   |                          |                                                      |   |   |   |   |   |
| C                                                   | -                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | -                                                   |                                     | -                        |                                                      |          |  |  |     |     |       |                    |                    |          |   |         |         |           |   |   |        |   |                     |                     |   |                          |                                                      |   |   |   |   |   |

1. If MCR[IDAM] field is equivalent to the format B only the fourteen most significant bits of the Identifier of the incoming frame are compared with the Rx FIFO filter.
2. If MCR[IDAM] field is equivalent to the format C only the eight most significant bits of the Identifier of the incoming frame are compared with the Rx FIFO filter.

37.4.18 Rx FIFO Information Register (CAN\_RXFIR)

RXFIR provides information on Rx FIFO.

This register is the port through which the CPU accesses the output of the RXFIR FIFO located in RAM. The RXFIR FIFO is written by the FlexCAN whenever a new message is moved into the Rx FIFO as well as its output is updated whenever the output of the Rx FIFO is updated with the next message. See Section "Rx FIFO" for instructions on reading this register.

Address: 0h base + 4Ch offset = 4Ch

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    |    |    |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|----|----|----|----|----|----|----|----|----|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22    | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | IDHIT |    |    |    |    |    |    |    |    |    |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    |    |    |
| Reset | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x*    | x* | x* | x* | x* | x* | x* | x* | x* |    |

- \* Notes:
- x = Undefined at reset.

CAN\_RXFIR field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                             |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–22<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                 |
| 23–31<br>IDHIT   | Identifier Acceptance Filter Hit Indicator<br><br>This field indicates which Identifier Acceptance Filter was hit by the received message that is in the output of the Rx FIFO. If multiple filters match the incoming message ID then the first matching IDAF found (lowest number) by the matching process is indicated. This field is valid only while the IFLAG[BUF5I] is asserted. |

37.4.19 Rx Individual Mask Registers (CAN\_RXIMRn)

These registers are located in RAM.

RXIMR are used as acceptance masks for ID filtering in Rx MBs and the Rx FIFO. If the Rx FIFO is not enabled, one mask register is provided for each available Mailbox, providing ID masking capability on a per Mailbox basis.

When the Rx FIFO is enabled (MCR[RFEN] bit is asserted), up to 32 Rx Individual Mask Registers can apply to the Rx FIFO ID Filter Table elements on a one-to-one correspondence depending on the setting of CTRL2[RFFN].

RXIMR can only be written by the CPU while the module is in Freeze mode; otherwise, they are blocked by hardware.

The Individual Rx Mask Registers are not affected by reset and must be explicitly initialized prior to any reception.

Address: 0h base + 880h offset + (4d × i), where i=0d to 63d

|       |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0        | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | MI[31:0] |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*       | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |

\* Notes:

- x = Undefined at reset.

### CAN\_RXIMRn field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>MI[31:0] | <p>Individual Mask Bits</p> <p>Each Individual Mask Bit masks the corresponding bit in both the Mailbox filter and Rx FIFO ID Filter Table element in distinct ways.</p> <p>For Mailbox filters, see the RXMGMASK register description.</p> <p>For Rx FIFO ID Filter Table elements, see the RXFGMASK register description.</p> <p>0 The corresponding bit in the filter is "don't care."</p> <p>1 The corresponding bit in the filter is checked.</p> |

## 37.4.20 Memory Error Control Register (CAN\_MECR)

This register contains control bits for memory error detection and correction (ECC).

### NOTE

When bit CTRL2[ECRWRE] is 0, this register is read-only and writes to this register result in bus transfer errors.

Address: 0h base + AE0h offset = AE0h

|       |                                               |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|-----------------------------------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0                                             | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | <div> <div>ECRWRDIS</div> <div>0</div> </div> |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |                                               |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 1                                             | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 1  | 1  | 0  | 0  |

## Memory map/register definition

|       |         |         |         |    |    |    |         |        |          |    |    |    |    |    |    |    |
|-------|---------|---------|---------|----|----|----|---------|--------|----------|----|----|----|----|----|----|----|
| Bit   | 16      | 17      | 18      | 19 | 20 | 21 | 22      | 23     | 24       | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | HAERRIE | FAERRIE | EXERRIE | 0  |    |    | RERRDIS | ECCDIS | NCEFAFRZ | 0  |    |    |    |    |    |    |
| W     |         |         |         |    |    |    |         |        |          |    |    |    |    |    |    |    |
| Reset | 0       | 0       | 0       | 0  | 0  | 0  | 0       | 0      | 1        | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

### CAN\_MECR field descriptions

| Field            | Description                                                                                                                                                                                                                                        |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>ECRWRDIS    | Error Configuration Register Write Disable<br>Disables writes on this register.<br><br>0 Write is enabled.<br>1 Write is disabled.                                                                                                                 |
| 1–11<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                            |
| 12<br>HANCEI_MSK | Host Access With Non-Correctable Errors Interrupt Mask<br>Enables the interrupt in case of non-correctable errors detected in memory reads issued by the host (CPU).<br><br>0 Interrupt is disabled.<br>1 Interrupt is enabled.                    |
| 13<br>FANCEI_MSK | FlexCAN Access With Non-Correctable Errors Interrupt Mask<br>Enables the interrupt in case of non-correctable errors detected in memory reads issued by the FlexCAN internal processes.<br><br>0 Interrupt is disabled.<br>1 Interrupt is enabled. |
| 14<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                            |
| 15<br>CEI_MSK    | Correctable Errors Interrupt Mask<br>Enables the interrupt in case of correctable errors detected in memory reads issued by the host or FlexCAN internal processes.<br><br>0 Interrupt is disabled.<br>1 Interrupt is enabled.                     |
| 16<br>HAERRIE    | Host Access Error Injection Enable<br>Enables the injection of errors only in memory reads issued by the host (CPU).<br><br>0 Injection is disabled.<br>1 Injection is enabled.                                                                    |
| 17<br>FAERRIE    | FlexCAN Access Error Injection Enable<br>Enables the injection of errors only in memory reads issued by the FlexCAN internal processes.<br><br>0 Injection is disabled.<br>1 Injection is enabled.                                                 |

Table continues on the next page...



**CAN\_MECR field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 18<br>EXTERRIE    | <p>Extended Error Injection Enable</p> <p>Memory accesses performed by internal FlexCAN processes are 64-bit. This bit extends the error injection on 32-bit memory accesses to the complementary 32-bit word using the same 32-bit error injection data and parity words. See Error Injection Data Pattern Register (FCERRIDPR) and Error Injection Parity Pattern Register (FCERRIPPR)</p> <p>0 Error injection is applied only to the 32-bit word.<br/>1 Error injection is applied to the 64-bit word.</p> |
| 19–21<br>Reserved | <p>This field is reserved.<br/>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                |
| 22<br>RERRDIS     | <p>Error Report Disable</p> <p>Disables the update of the error report registers. The update of error-related flags and the generation of bus transfer errors are still active.</p> <p><b>NOTE:</b> When reading the report registers, this bit must be cleared to assure coherence on the consecutive register reads.</p> <p>0 Enable updates of the error report registers.<br/>1 Disable updates of the error report registers.</p>                                                                         |
| 23<br>ECCDIS      | <p>Error Correction Disable</p> <p>Disables completely the memory error detection and correction mechanism. Besides disabling the error report mechanism, it also stops the update of the error-related flags and generation of bus transfer errors. The parity bits continue being calculated and written into memory on write transactions.</p> <p>0 Enable memory error correction.<br/>1 Disable memory error correction.</p>                                                                              |
| 24<br>NCEFAFRZ    | <p>Non-Correctable Errors In FlexCAN Access Put Device In Freeze Mode</p> <p>Determines the response when a non-correctable error is detected in a memory read performed by FlexCAN internal processes. In this case, entering Freeze mode prevents corrupted data from being treated as valid by FlexCAN internal processes.</p> <p>0 Keep normal operation.<br/>1 Put FlexCAN in Freeze mode (according to the "Freeze mode" section).</p>                                                                   |
| 25–31<br>Reserved | <p>This field is reserved.<br/>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                |

**37.4.21 Error Injection Address Register (CAN\_ERRIAR)**

This register holds the address where error is to be injected.

Use the following table to convert from the memory map address to the location in the physical FlexCAN RAM:

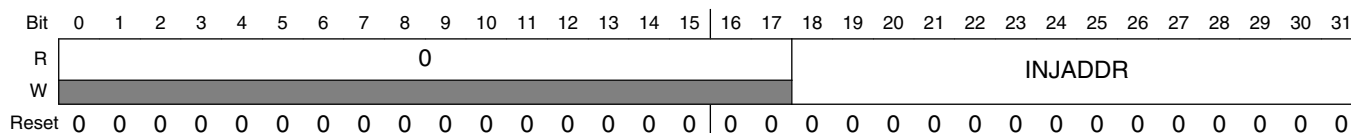
| RAM contents      | Injection address | Memory map |
|-------------------|-------------------|------------|
| FlexCAN Registers | Not mapped        | -          |

*Table continues on the next page...*

## Memory map/register definition

| RAM contents  | Injection address | Memory map |
|---------------|-------------------|------------|
| MBs           | 0x0000            | 0x0080     |
| Reserved      | -                 | 0x0480     |
| RXIMRs        | 0x0400            | 0x0880     |
| Reserved      | -                 | 0x0980     |
| RXFIR_0       | 0x0500            | 0x0A80     |
| RXFIR_1       | 0x0504            | 0x0A84     |
| RXFIR_2       | 0x0508            | 0x0A88     |
| RXFIR_3       | 0x050C            | 0x0A8C     |
| RXFIR_4       | 0x0510            | 0x0A90     |
| RXFIR_5       | 0x0514            | 0x0A94     |
| Reserved      | -                 | 0x0A98     |
| RXMGMASK      | 0x0520            | 0x0AA0     |
| RXFGMASK      | 0x0524            | 0x0AA4     |
| RX14MASK      | 0x0528            | 0x0AA8     |
| RX15MASK      | 0x052C            | 0x0AAC     |
| SMB_TX        | Not Mapped        | 0x0AB0     |
| SMB_RX0       | 0x0540            | 0x0AC0     |
| SMB_RX1       | 0x0550            | 0x0AD0     |
| ECC Registers | Not mapped        | 0x0AE0     |
| Reserved      | -                 | 0x0B00     |

Address: 0h base + AE4h offset = AE4h



### CAN\_ERRIAR field descriptions

| Field            | Description                                                                                                 |
|------------------|-------------------------------------------------------------------------------------------------------------|
| 0–17<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                     |
| 18–31<br>INJADDR | Address Where Error Is To Be Injected<br><br>The two least significant bits of this field always read as 0. |

## 37.4.22 Error Injection Data Pattern Register (CAN\_ERRIDPR)

Holds the error pattern to be injected in the data word read from memory.

Address: 0h base + AE8h offset = AE8h

|       |       |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0     | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | DFLIP |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |       |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0     | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**CAN\_ERRIDPR field descriptions**

| Field         | Description                                                                                                                     |
|---------------|---------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>DFLIP | Data flip pattern<br>Bits set to 1 in the flip pattern cause the corresponding data bit in the word read from memory to invert. |

## 37.4.23 Error Injection Parity Pattern Register (CAN\_ERRIPPR)

Holds the error pattern to be injected in parity bits read from memory along with data word.

Bits set to 1 in the flip pattern cause the corresponding parity bit in the word read from memory to invert.

Address: 0h base + AECh offset = AECh

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   | 0 |   |    |    |    |    |    |    | 0  |    |    |    |    |    |    |    | 0  |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**CAN\_ERRIPPR field descriptions**

| Field             | Description                                                                             |
|-------------------|-----------------------------------------------------------------------------------------|
| 0–2<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 3–7<br>PFLIP3     | Parity Flip Pattern For Byte 3 (most significant)                                       |
| 8–10<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 11–15<br>PFLIP2   | Parity Flip Pattern For Byte 2                                                          |
| 16–18<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |

*Table continues on the next page...*

**CAN\_ERRIPPR field descriptions (continued)**

| Field             | Description                                                                             |
|-------------------|-----------------------------------------------------------------------------------------|
| 19–23<br>PFLIP1   | Parity Flip Pattern For Byte 1                                                          |
| 24–26<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 27–31<br>PFLIP0   | Parity Flip Pattern For Byte 0 (Least Significant)                                      |

**37.4.24 Error Report Address Register (CAN\_RERRAR)**

Reports the address used for an access in which an error (correctable or non-correctable) was detected, and also reports the identification of the source of that access.

This address is always reported using a 32-bit alignment. Non-aligned accesses (ERRADDR[1:0] non-0) are reported with the address aligned and data is reported in RERRDR accordingly shifted. In case of errors detected in accesses larger than 32-bit (as performed by FlexCAN internal processes), the address of the 32-bit word in which the error was detected is reported. In case of errors detected in more than one 32-bit word, only the least significant address is reported.

Address: 0h base + AF0h offset = AF0h

|       |   |   |   |   |   |   |   |     |   |   |    |    |    |    |      |    |  |
|-------|---|---|---|---|---|---|---|-----|---|---|----|----|----|----|------|----|--|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7   | 8 | 9 | 10 | 11 | 12 | 13 | 14   | 15 |  |
| R     | 0 |   |   |   |   |   |   | NCE | 0 |   |    |    |    |    | SAID |    |  |
| W     |   |   |   |   |   |   |   |     |   |   |    |    |    |    |      |    |  |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0 | 0 | 0  | 0  | 0  | 0  | 0    | 0  |  |

|       |    |    |         |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----|----|---------|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 16 | 17 | 18      | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0  |    | ERRADDR |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |    |    |         |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0  | 0  | 0       | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**CAN\_RERRAR field descriptions**

| Field           | Description                                                                                                                                                              |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–6<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                  |
| 7<br>NCE        | Non-Correctable Error<br><br>Indicates that the report is due to an non-correctable error.<br><br>0 Reporting a correctable-error<br>1 Reporting a non-correctable error |

*Table continues on the next page...*

**CAN\_RERRAR field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |           |                 |   |         |   |         |   |                |   |          |   |         |     |          |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|-----------------|---|---------|---|---------|---|----------------|---|----------|---|---------|-----|----------|
| 8–12<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |           |                 |   |         |   |         |   |                |   |          |   |         |     |          |
| 13–15<br>SAID     | <p>SAID[2] — Identification of the requestor of the memory read request:</p> <ul style="list-style-type: none"> <li>0 = Requested by FlexCAN internal processes</li> <li>1 = Requested by Host (CPU)</li> </ul> <p>SAID[1] — Details of FlexCAN operation:</p> <ul style="list-style-type: none"> <li>0 = Move</li> <li>1 = Scanning</li> </ul> <p>SAID[0] — Operation that requested the memory read:</p> <ul style="list-style-type: none"> <li>0 = Transmission</li> <li>1 = Reception</li> </ul> <p style="text-align: center;"><b>Table 37-91. Source of memory access</b></p> <table> <tr> <th>SAID[2:0]</th><th>Error during...</th></tr> <tr> <td>0</td><td>Tx Move</td></tr> <tr> <td>1</td><td>Rx Move</td></tr> <tr> <td>2</td><td>Tx Arbitration</td></tr> <tr> <td>3</td><td>Rx Match</td></tr> <tr> <td>4</td><td>Tx Move</td></tr> <tr> <td>5-7</td><td>Reserved</td></tr> </table> | SAID[2:0] | Error during... | 0 | Tx Move | 1 | Rx Move | 2 | Tx Arbitration | 3 | Rx Match | 4 | Tx Move | 5-7 | Reserved |
| SAID[2:0]         | Error during...                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |           |                 |   |         |   |         |   |                |   |          |   |         |     |          |
| 0                 | Tx Move                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |           |                 |   |         |   |         |   |                |   |          |   |         |     |          |
| 1                 | Rx Move                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |           |                 |   |         |   |         |   |                |   |          |   |         |     |          |
| 2                 | Tx Arbitration                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |           |                 |   |         |   |         |   |                |   |          |   |         |     |          |
| 3                 | Rx Match                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |           |                 |   |         |   |         |   |                |   |          |   |         |     |          |
| 4                 | Tx Move                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |           |                 |   |         |   |         |   |                |   |          |   |         |     |          |
| 5-7               | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |           |                 |   |         |   |         |   |                |   |          |   |         |     |          |
| 16–17<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |           |                 |   |         |   |         |   |                |   |          |   |         |     |          |
| 18–31<br>ERRADDR  | <p>Address Where The Error Was Detected</p> <p>See the description of the Error Injection Address Register (ERRIAR).</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |           |                 |   |         |   |         |   |                |   |          |   |         |     |          |

**37.4.25 Error Report Data Register (CAN\_RERRDR)**

Reports the raw data (unmodified by the correction performed by ECC logic) read from memory with error. The value reported does not represent the transient values of the BUSY bit (see the “Message Buffer Code for Rx buffers” table) when reading a Message Buffer.

Address: 0h base + AF4h offset = AF4h

|       |       |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0     | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | RDATA |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |       |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0     | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**CAN\_RERRDR field descriptions**

| Field         | Description                               |
|---------------|-------------------------------------------|
| 0–31<br>RDATA | Raw data word read from memory with error |

**37.4.26 Error Report Syndrome Register (CAN\_RERRSYNR)**

Holds the syndrome detected in a memory read with error. It also reports the bytes which were read in this 32-bit read transaction.

Each SYNDn field indicates the type of error and which bit in byte (n) is affected by the error. (SYND3 corresponds to the most significant byte in the data word read from memory; SYND0 corresponds to the least significant.)

**Table 37-94. Syndrome Definition**

| SYNDn (hex) | Type | Bit affected          |
|-------------|------|-----------------------|
| 00          | -    | none (no error)       |
| 01          | Code | 0                     |
| 02          | Code | 1                     |
| 04          | Code | 2                     |
| 07          | Data | 5                     |
| 08          | Code | 3                     |
| 0E          | Data | 7                     |
| 10          | Code | 4                     |
| 13          | Data | 2                     |
| 15          | Data | 6                     |
| 16          | Data | 1                     |
| 19          | Data | 3                     |
| 1A          | Data | 4                     |
| 1C          | Data | 0                     |
| All others  | -    | Non-correctable error |

Each BEN field indicates which byte in the 32-bit word reported was effectively read. The syndrome bits are calculated for all bytes, even for the non-read ones. Errors detected in non-read bytes are indicated (see the "Error Indication" section) and reported (see the "Error Reporting" section).

Address: 0h base + AF8h offset = AF8h

|       |     |   |   |   |   |   |   |   |     |   |    |    |    |    |    |    |
|-------|-----|---|---|---|---|---|---|---|-----|---|----|----|----|----|----|----|
| Bit   | 0   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8   | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | BE3 | 0 |   |   |   |   |   |   | BE2 | 0 |    |    |    |    |    |    |
| W     |     |   |   |   |   |   |   |   |     |   |    |    |    |    |    |    |
| Reset | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |     |    |    |    |    |    |    |    |     |    |    |    |    |    |    |    |
|-------|-----|----|----|----|----|----|----|----|-----|----|----|----|----|----|----|----|
| Bit   | 16  | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24  | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | BE1 | 0  |    |    |    |    |    |    | BE0 | 0  |    |    |    |    |    |    |
| W     |     |    |    |    |    |    |    |    |     |    |    |    |    |    |    |    |
| Reset | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**CAN\_RERRSYNR field descriptions**

| Field             | Description                                                                                     |
|-------------------|-------------------------------------------------------------------------------------------------|
| 0<br>BE3          | Byte Enabled For Byte 3 (Most Significant)<br>0 The byte was not read.<br>1 The byte was read.  |
| 1–2<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.         |
| 3–7<br>SYND3      | Error Syndrome For Byte 3 (Most Significant)<br>See the "Syndrome Definition" table.            |
| 8<br>BE2          | Byte Enabled For Byte 2<br>0 The byte was not read.<br>1 The byte was read.                     |
| 9–10<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.         |
| 11–15<br>SYND2    | Error Syndrome For Byte 2<br>See the "Syndrome Definition" table.                               |
| 16<br>BE1         | Byte Enabled For Byte 1<br>0 The byte was not read.<br>1 The byte was read.                     |
| 17–18<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.         |
| 19–23<br>SYND1    | Error Syndrome for Byte 1<br>See the "Syndrome Definition" table.                               |
| 24<br>BE0         | Byte Enabled For Byte 0 (least significant)<br>0 The byte was not read.<br>1 The byte was read. |
| 25–26<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.         |
| 27–31<br>SYND0    | Error Syndrome For Byte 0 (least significant)<br>See the "Syndrome Definition" table.           |

37.4.27 Error Status Register (CAN\_ERRSR)

Holds the status bits of the error correction and detection operations. After raised, these flags must be cleared by writing 1 to them. Writing 0 has no effect.

Address: 0h base + AFCh offset = AFCh

|       |   |   |   |   |   |   |   |   |   |   |    |    |         |         |    |      |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|---------|---------|----|------|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12      | 13      | 14 | 15   |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    | HANCEIF | FANCEIF | 0  | CEIF |
| W     |   |   |   |   |   |   |   |   |   |   |    |    | w1c     | w1c     |    | w1c  |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0       | 0       | 0  | 0    |

|       |    |    |    |    |    |    |    |    |    |    |    |    |          |          |    |       |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----------|----------|----|-------|
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28       | 29       | 30 | 31    |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    | HANCEIOF | FANCEIOF | 0  | CEIOF |
| W     |    |    |    |    |    |    |    |    |    |    |    |    | w1c      | w1c      |    | w1c   |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0        | 0        | 0  | 0     |

CAN\_ERRSR field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                           |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–11<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                               |
| 12<br>HANCEIF    | Host Access With Non-Correctable Error Interrupt Flag<br><br>Indicates that a non-correctable error was detected in a memory access initiated by Host. A bus transfer error is asserted for that access. If MECR[HANCEI_MSK] is set, the interrupt is asserted.<br><br>0 No non-correctable errors were detected in Host accesses so far.<br>1 A non-correctable error was detected in a Host access. |
| 13<br>FANCEIF    | FlexCAN Access With Non-Correctable Error Interrupt Flag                                                                                                                                                                                                                                                                                                                                              |

Table continues on the next page...



**CAN\_ERRSR field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | Indicates that a non-correctable error was detected in a memory access initiated by FlexCAN internal processes. If MECR[FANCEI_MSK] is set, the interrupt is asserted.<br><br>0 No non-correctable errors were detected in FlexCAN accesses so far.<br>1 A non-correctable error was detected in a FlexCAN access.                                                                                                   |
| 14<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                              |
| 15<br>CEIF        | Correctable Error Interrupt Flag<br><br>Indicates that a correctable error was detected in a memory access. If MECR[CEI_MSK] is set, the interrupt is asserted.<br><br>0 No correctable errors were detected so far.<br>1 A correctable error was detected.                                                                                                                                                          |
| 16–27<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                              |
| 28<br>HANCEIOF    | Host Access With Non-Correctable Error Interrupt Overrun Flag<br><br>Indicates that a non-correctable error was detected in a memory access initiated by Host when HANCEIF was set. No interrupt is associated to this flag. (See the "Error Indication" section.)<br><br>0 No overrun on non-correctable errors in Host access<br>1 Overrun on non-correctable errors in Host access                                |
| 29<br>FANCEIOF    | FlexCAN Access With Non-Correctable Error Interrupt Overrun Flag<br><br>Indicates that a non-correctable error was detected in a memory access initiated by FlexCAN internal processes when FANCEIF was set. No interrupt is associated to this flag. (See the "Error Indication" section.)<br><br>0 No overrun on non-correctable errors in FlexCAN access<br>1 Overrun on non-correctable errors in FlexCAN access |
| 30<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                              |
| 31<br>CEIOF       | Correctable Error Interrupt Overrun Flag<br><br>Indicates that a correctable error was detected in a memory access when CEIF was set. No interrupt is associated to this flag. (See the "Error Indication" section.)<br><br>0 No overrun on correctable errors<br>1 Overrun on correctable errors                                                                                                                    |

**37.4.28 Message buffer structure**

The message buffer structure used by the FlexCAN module is represented in the following figure. Both Extended (29-bit identifier) and Standard (11-bit identifier) frames used in the CAN specification (Version 2.0 Part B) are represented. Each individual MB is formed by 16 bytes.

The memory area from 0x80 to 0x47F is used by the mailboxes.

**Table 37-96. Message buffer structure**

|     | 0           | 1 | 2 | 3                      | 4                           | 7 | 8           | 9   | 10  | 11  | 12  | 13 | 14          | 15            | 16         | 23          | 24 | 31 |  |
|-----|-------------|---|---|------------------------|-----------------------------|---|-------------|-----|-----|-----|-----|----|-------------|---------------|------------|-------------|----|----|--|
| 0x0 |             |   |   |                        | CODE                        |   |             | SRR | IDE | RTR | DLC |    |             |               | TIME STAMP |             |    |    |  |
| 0x4 | PRIO        |   |   | ID (Standard/Extended) |                             |   |             |     |     |     |     |    |             | ID (Extended) |            |             |    |    |  |
| 0x8 | Data Byte 0 |   |   |                        |                             |   | Data Byte 1 |     |     |     |     |    | Data Byte 2 |               |            | Data Byte 3 |    |    |  |
| 0xC | Data Byte 4 |   |   |                        |                             |   | Data Byte 5 |     |     |     |     |    | Data Byte 6 |               |            | Data Byte 7 |    |    |  |
|     |             |   |   |                        | = Unimplemented or Reserved |   |             |     |     |     |     |    |             |               |            |             |    |    |  |

**CODE** — Message Buffer Code

This 4-bit field can be accessed (read or write) by the CPU and by the FlexCAN module itself, as part of the message buffer matching and arbitration process. The encoding is shown in [Table 37-97](#) and [Table 37-98](#). See [Functional description](#) for additional information.

**Table 37-97. Message buffer code for Rx buffers**

| CODE description                        | Rx code BEFORE receive new frame | SRV <sup>1</sup> | Rx code AFTER successful reception <sup>2</sup> | RRS <sup>3</sup> | Comment                                                                                                                                                                                                                                                                                               |
|-----------------------------------------|----------------------------------|------------------|-------------------------------------------------|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0b0000: INACTIVE — MB is not active.    | INACTIVE                         | —                | —                                               | —                | MB does not participate in the matching process.                                                                                                                                                                                                                                                      |
| 0b0100: EMPTY — MB is active and empty. | EMPTY                            | —                | FULL                                            | —                | When a frame is received successfully (after the <a href="#">Move-in</a> process), the CODE field is automatically updated to FULL.                                                                                                                                                                   |
| 0b0010: FULL — MB is full.              | FULL                             | Yes              | FULL                                            | —                | The act of reading the C/S word followed by unlocking the MB (SRV) does not make the code return to EMPTY. It remains FULL. If a new frame is moved to the MB after the MB was serviced, the code still remains FULL. See <a href="#">Matching process</a> for matching details related to FULL code. |

Table continues on the next page...

**Table 37-97. Message buffer code for Rx buffers (continued)**

| CODE description                                                                                                                   | Rx code BEFORE receive new frame | SRV <sup>1</sup> | Rx code AFTER successful reception <sup>2</sup> | RRS <sup>3</sup> | Comment                                                                                                                                                                                                                                                                                                           |
|------------------------------------------------------------------------------------------------------------------------------------|----------------------------------|------------------|-------------------------------------------------|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                                                                                                    |                                  | No               | OVERRUN                                         | —                | If the MB is FULL and a new frame is moved to this MB before the CPU services it, the CODE field is automatically updated to OVERRUN. See <a href="#">Matching process</a> for details about overrun behavior.                                                                                                    |
| 0b0110: OVERRUN — MB is being overwritten into a full buffer.                                                                      | OVERRUN                          | Yes              | FULL                                            | —                | If the CODE field indicates OVERRUN and CPU has serviced the MB, when a new frame is moved to the MB then the code returns to FULL.                                                                                                                                                                               |
|                                                                                                                                    |                                  | No               | OVERRUN                                         | —                | If the CODE field already indicates OVERRUN, and another new frame must be moved, the MB will be overwritten again, and the code will remain OVERRUN. See <a href="#">Matching process</a> for details about overrun behavior.                                                                                    |
| 0b1010: RANSWER <sup>4</sup> — A frame was configured to recognize a Remote Request Frame and transmit a Response Frame in return. | RANSWER                          | —                | TANSWER(0b1110)                                 | 0                | A Remote Answer was configured to recognize a remote request frame received. After that an MB is set to transmit a response frame. The code is automatically changed to TANSWER (0b1110). See <a href="#">Matching process</a> for details. If CTRL2[RRS] is negated, transmit a response frame whenever a remote |

Table continues on the next page...

**Table 37-97. Message buffer code for Rx buffers (continued)**

| CODE description                                                                                | Rx code BEFORE receive new frame | SRV <sup>1</sup> | Rx code AFTER successful reception <sup>2</sup> | RRS <sup>3</sup> | Comment                                                                                                             |
|-------------------------------------------------------------------------------------------------|----------------------------------|------------------|-------------------------------------------------|------------------|---------------------------------------------------------------------------------------------------------------------|
|                                                                                                 |                                  |                  |                                                 |                  | request frame with the same ID is received.                                                                         |
|                                                                                                 |                                  | —                | —                                               | 1                | This code is ignored during matching and arbitration process. See <a href="#">Matching process</a> for details.     |
| CODE[0]=1b1: BUSY — FlexCAN is updating the contents of the MB. The CPU must not access the MB. | BUSY <sup>5</sup>                | —                | FULL                                            | —                | Indicates that the MB is being updated. It will be negated automatically and does not interfere with the next CODE. |
|                                                                                                 |                                  | —                | OVERRUN                                         | —                |                                                                                                                     |

1. SRV: Serviced MB. MB was read and unlocked by reading TIMER or other MB.
2. A frame is considered a successful reception after the frame to be moved to MB (move-in process). See [Move-in](#) for details.
3. Remote Request Stored bit from CTRL2 register. See section "Control 2 Register (CTRL2)" for details.
4. Code 0b1010 is not considered Tx and an MB with this code should not be aborted.
5. Note that for Tx MBs, the BUSY bit should be ignored upon read, except when AEN bit is set in the MCR register. If this bit is asserted, the corresponding MB does not participate in the matching process.

**Table 37-98. Message buffer code for Tx buffers**

| CODE Description                                                    | Tx Code BEFORE tx frame | MB RTR | Tx Code AFTER successful transmission | Comment                                                                                                                               |
|---------------------------------------------------------------------|-------------------------|--------|---------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| 0b1000: INACTIVE — MB is not active                                 | INACTIVE                | —      | —                                     | MB does not participate in arbitration process.                                                                                       |
| 0b1001: ABORT — MB is aborted                                       | ABORT                   | —      | —                                     | MB does not participate in arbitration process.                                                                                       |
| 0b1100: DATA — MB is a Tx Data Frame (MB RTR must be 0)             | DATA                    | 0      | INACTIVE                              | Transmit data frame unconditionally once. After transmission, the MB automatically returns to the INACTIVE state.                     |
| 0b1100: REMOTE — MB is a Tx Remote Request Frame (MB RTR must be 1) | REMOTE                  | 1      | EMPTY                                 | Transmit remote request frame unconditionally once. After transmission, the MB automatically becomes an Rx Empty MB with the same ID. |
| 0b1110: TANSWER — MB is a Tx Response                               | TANSWER                 | —      | RANSWER                               | This is an intermediate code that is                                                                                                  |

**Table 37-98. Message buffer code for Tx buffers**

| CODE Description                            | Tx Code BEFORE tx frame | MB RTR | Tx Code AFTER successful transmission | Comment                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|---------------------------------------------|-------------------------|--------|---------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Frame from an incoming Remote Request Frame |                         |        |                                       | automatically written to the MB by the CHI as a result of a match to a remote request frame. The remote response frame will be transmitted unconditionally once, and then the code will automatically return to RANSWER (0b1010). The CPU can also write this code with the same effect. The remote response frame can be either a data frame or another remote request frame depending on the RTR bit value. See <a href="#">Matching process</a> and <a href="#">Arbitration process</a> for details. |

**SRR** — Substitute Remote Request

Fixed recessive bit, used only in extended format. It must be set to one by the user for transmission (Tx Buffers) and will be stored with the value received on the CAN bus for Rx receiving buffers. It can be received as either recessive or dominant. If FlexCAN receives this bit as dominant, then it is interpreted as an arbitration loss.

1 = Recessive value is compulsory for transmission in extended format frames

0 = Dominant is not a valid value for transmission in extended format frames

**IDE** — ID Extended Bit

This field identifies whether the frame format is standard or extended.

1 = Frame format is extended

0 = Frame format is standard

**RTR** — Remote Transmission Request

This bit affects the behavior of remote frames and is part of the reception filter. See [Table 37-97](#), [Table 37-98](#), and the description of the RRS bit in Control 2 Register (CTRL2) for additional details.

If FlexCAN transmits this bit as '1' (recessive) and receives it as '0' (dominant), it is interpreted as an arbitration loss. If this bit is transmitted as '0' (dominant), then if it is received as '1' (recessive), the FlexCAN module treats it as a bit error. If the value received matches the value transmitted, it is considered a successful bit transmission.

1 = Indicates the current MB may have a remote request frame to be transmitted if MB is Tx. If the MB is Rx then incoming remote request frames may be stored.

0 = Indicates the current MB has a data frame to be transmitted. In Rx MB it may be considered in matching processes.

### **DLC** — Length of Data in Bytes

This 4-bit field is the length (in bytes) of the Rx or Tx data, which is located in offset 0x8 through 0xF of the MB space (see the [Table 37-96](#)). In reception, this field is written by the FlexCAN module, copied from the DLC (Data Length Code) field of the received frame. In transmission, this field is written by the CPU and corresponds to the DLC field value of the frame to be transmitted. When RTR = 1, the frame to be transmitted is a remote frame and does not include the data field, regardless of the DLC field (see [Table 37-99](#)).

### **TIME STAMP** — Free-Running Counter Time Stamp

This 16-bit field is a copy of the Free-Running Timer, captured for Tx and Rx frames at the time when the beginning of the Identifier field appears on the CAN bus.

### **PRI0** — Local priority

This 3-bit field is used only when LPRI0\_EN bit is set in MCR, and it only makes sense for Tx mailboxes. These bits are not transmitted. They are appended to the regular ID to define the transmission priority. See [Arbitration process](#).

### **ID** — Frame Identifier

In standard frame format, only the 11 most significant bits (3 to 13) are used for frame identification in both receive and transmit cases. The 18 least significant bits are ignored. In extended frame format, all bits are used for frame identification in both receive and transmit cases.

### **DATA BYTE 0–7** — Data Field

Up to eight bytes can be used for a data frame.

For Rx frames, the data is stored as it is received from the CAN bus. DATA BYTE (*n*) is valid only if *n* is less than DLC as shown in the table below.

For Tx frames, the CPU prepares the data field to be transmitted within the frame.

**Table 37-99. DATA BYTES validity**

| DLC | Valid DATA BYTES |
|-----|------------------|
| 0   | none             |
| 1   | DATA BYTE 0      |
| 2   | DATA BYTE 0–1    |
| 3   | DATA BYTE 0–2    |
| 4   | DATA BYTE 0–3    |
| 5   | DATA BYTE 0–4    |
| 6   | DATA BYTE 0–5    |
| 7   | DATA BYTE 0–6    |
| 8   | DATA BYTE 0–7    |

### 37.4.29 Rx FIFO structure

When the MCR[RFEN] bit is set, the memory area from 0x80 to 0xDC (which is normally occupied by MBs 0–5) is used by the reception FIFO engine.

The region 0x80-0x8C contains the output of the FIFO which must be read by the CPU as a message buffer. This output contains the oldest message that has been received but not yet read. The region 0x90-0xDC is reserved for internal use of the FIFO engine.

An additional memory area, which starts at 0xE0 and may extend up to 0x2DC (normally occupied by MBs 6–37) depending on the CTRL2[RFFN] field setting, contains the ID filter table (configurable from 8 to 128 table elements) that specifies filtering criteria for accepting frames into the FIFO.

Out of reset, the ID filter table flexible memory area defaults to 0xE0 and extends only to 0xFC, which corresponds to MBs 6 to 7 for RFFN = 0, for backward compatibility with previous versions of FlexCAN.

The following shows the Rx FIFO data structure.

**Table 37-100. Rx FIFO structure**

|      | 0           | 3 | 7           | 8           | 9   | 10  | 11  | 12  | 13 | 14 | 15          | 16         | 23          | 24 | 31          |  |
|------|-------------|---|-------------|-------------|-----|-----|-----|-----|----|----|-------------|------------|-------------|----|-------------|--|
| 0x80 |             |   |             |             | SRR | IDE | RTR | DLC |    |    |             | TIME STAMP |             |    |             |  |
| 0x84 |             |   | ID standard |             |     |     |     |     |    |    | ID extended |            |             |    |             |  |
| 0x88 | Data byte 0 |   |             | Data byte 1 |     |     |     |     |    |    |             |            | Data byte 2 |    | Data byte 3 |  |
| 0x8C | Data byte 4 |   |             | Data byte 5 |     |     |     |     |    |    |             |            | Data byte 6 |    | Data byte 7 |  |
| 0x90 | Reserved    |   |             |             |     |     |     |     |    |    |             |            |             |    |             |  |
| to   |             |   |             |             |     |     |     |     |    |    |             |            |             |    |             |  |
| 0xDC |             |   |             |             |     |     |     |     |    |    |             |            |             |    |             |  |

*Table continues on the next page...*

**Table 37-100. Rx FIFO structure (continued)**

|                     |                                   |
|---------------------|-----------------------------------|
| 0xE0                | ID filter table element 0         |
| 0xE4                | ID filter table element 1         |
| 0xE8<br>to<br>0x2D4 | ID filter table elements 2 to 125 |
| 0x2D8               | ID filter table element 126       |
| 0x2DC               | ID filter table element 127       |
|                     | = Unimplemented or reserved       |

Each ID filter table element occupies an entire 32-bit word and can be compounded by one, two, or four Identifier Acceptance Filters (IDAF) depending on the MCR[IDAM] field setting. The following figures show the IDAF indexation.

The following table shows the three different formats of the ID table elements. Note that all elements of the table must have the same format. See [Rx FIFO](#) for more information.

**Table 37-101. ID table structure**

| Format | 0                           | 1   | 2                                             | 7 | 8                           | 15 | 16  | 17  | 18                                              | 23 | 24 | 30 | 31                           |
|--------|-----------------------------|-----|-----------------------------------------------|---|-----------------------------|----|-----|-----|-------------------------------------------------|----|----|----|------------------------------|
| A      | RTR                         | IDE | RXIDA<br>(standard = 2–12, extended = 2–30)   |   |                             |    |     |     |                                                 |    |    |    |                              |
| B      | RTR                         | IDE | RXIDB_0<br>(standard = 2–12, extended = 2–15) |   |                             |    | RTR | IDE | RXIDB_1<br>(standard = 18–28, extended = 18–31) |    |    |    |                              |
| C      | RXIDC_0<br>(std/ext = 0–7)  |     |                                               |   | RXIDC_1<br>(std/ext = 8–15) |    |     |     | RXIDC_2<br>(std/ext = 16–23)                    |    |    |    | RXIDC_3<br>(std/ext = 24–31) |
|        | = Unimplemented or Reserved |     |                                               |   |                             |    |     |     |                                                 |    |    |    |                              |

### **RTR** — Remote Frame

This bit specifies if Remote Frames are accepted into the FIFO if they match the target ID.

1 = Remote Frames can be accepted and data frames are rejected

0 = Remote Frames are rejected and data frames can be accepted

### **IDE** — Extended Frame

Specifies whether extended or standard frames are accepted into the FIFO if they match the target ID.

1 = Extended frames can be accepted and standard frames are rejected



0 = Extended frames are rejected and standard frames can be accepted

### **RXIDA** — Rx Frame Identifier (Format A)

Specifies an ID to be used as acceptance criteria for the FIFO. In the standard frame format, only the 11 most significant bits (2 to 12) are used for frame identification. In the extended frame format, all bits are used.

### **RXIDB\_0, RXIDB\_1** — Rx Frame Identifier (Format B)

Specifies an ID to be used as acceptance criteria for the FIFO. In the standard frame format, the 11 most significant bits (a full standard ID) (2 to 12 and 18 to 28) are used for frame identification. In the extended frame format, all 14 bits of the field are compared to the 14 most significant bits of the received ID.

### **RXIDC\_0, RXIDC\_1, RXIDC\_2, RXIDC\_3** — Rx Frame Identifier (Format C)

Specifies an ID to be used as acceptance criteria for the FIFO. In both standard and extended frame formats, all 8 bits of the field are compared to the 8 most significant bits of the received ID.

## **37.5 Functional description**

The FlexCAN module is a CAN protocol engine with a very flexible mailbox system for transmitting and receiving CAN frames. The mailbox system is composed by a set of Message Buffers (MB) that store configuration and control data, time stamp, message ID and data (see [Message buffer structure](#)). The memory corresponding to the first 38 MBs can be configured to support a FIFO reception scheme with a powerful ID filtering mechanism, capable of checking incoming frames against a table of IDs (up to 128 extended IDs or 256 standard IDs or 512 8-bit ID slices), with individual mask register for up to 32 ID Filter Table elements. Simultaneous reception through FIFO and mailbox is supported. For mailbox reception, a matching algorithm makes it possible to store received frames only into MBs that have the same ID programmed on its ID field. A masking scheme makes it possible to match the ID programmed on the MB with a range of IDs on received CAN frames. For transmission, an arbitration algorithm decides the prioritization of MBs to be transmitted based on the message ID (optionally augmented by 3 local priority bits) or the MB ordering.

Before proceeding with the functional description, an important concept must be explained. A Message Buffer is said to be "active" at a given time if it can participate in both the Matching and Arbitration processes. An Rx MB with a 0b0000 code is inactive (refer to [Table 37-97](#)). Similarly, a Tx MB with a 0b1000 or 0b1001 code is also inactive (refer to [Table 37-98](#)).

### 37.5.1 Transmit process

To transmit a CAN frame, the CPU must prepare a Message Buffer for transmission by executing the following procedure:

1. Check whether the respective interrupt bit is set and clear it.
2. If the MB is active (transmission pending), write the ABORT code (0b1001) to the CODE field of the Control and Status word to request an abortion of the transmission. Wait for the corresponding IFLAG to be asserted by polling the IFLAG register or by the interrupt request if enabled by the respective IMASK. Then read back the CODE field to check if the transmission was aborted or transmitted (see [Transmission abort mechanism](#)). If backwards compatibility is desired (MCR[AEN] bit is negated), just write the INACTIVE code (0b1000) to the CODE field to inactivate the MB but then the pending frame may be transmitted without notification (see [Mailbox inactivation](#)).
3. Write the ID word.
4. Write the data bytes.
5. Write the DLC, Control, and CODE fields of the Control and Status word to activate the MB.

When the MB is activated, it will participate into the arbitration process and eventually be transmitted according to its priority. At the end of the successful transmission, the value of the Free Running Timer is written into the Time Stamp field, the CODE field in the Control and Status word is updated, the CRC Register is updated, a status flag is set in the Interrupt Flag Register and an interrupt is generated if allowed by the corresponding Interrupt Mask Register bit. The new CODE field after transmission depends on the code that was used to activate the MB (see [Table 37-97](#) and [Table 37-98](#) in [Message buffer structure](#)).

When the Abort feature is enabled (MCR[AEN] is asserted), after the Interrupt Flag is asserted for a Mailbox configured as transmit buffer, the Mailbox is blocked, therefore the CPU is not able to update it until the Interrupt Flag is negated by CPU. This means that the CPU must clear the corresponding IFLAG before starting to prepare this MB for a new transmission or reception.

## 37.5.2 Arbitration process

The arbitration process scans the Mailboxes searching the Tx one that holds the message to be sent in the next opportunity. This Mailbox is called the *arbitration winner*.

The scan starts from the lowest number Mailbox and runs toward the higher ones.

The arbitration process is triggered in the following events:

- From the CRC field of the CAN frame. The start point depends on the CTRL2[TASD] field value.
- During the Error Delimiter field of a CAN frame.
- During the Overload Delimiter field of a CAN frame.
- When the winner is inactivated and the CAN bus has still not reached the first bit of the Intermission field.
- When there is CPU write to the C/S word of a winner MB and the CAN bus has still not reached the first bit of the Intermission field.
- When CHI is in Idle state and the CPU writes to the C/S word of any MB.
- When FlexCAN exits Bus Off state.
- Upon leaving Freeze mode or Low Power mode.

If the arbitration process does not manage to evaluate all Mailboxes before the CAN bus has reached the first bit of the Intermission field the temporary arbitration winner is invalidated and the FlexCAN will not compete for the CAN bus in the next opportunity.

The arbitration process selects the winner among the active Tx Mailboxes at the end of the scan according to both CTRL1[LBUF] and MCR[LPRIOEN] bits settings.

### 37.5.2.1 Lowest-number Mailbox first

If CTRL1[LBUF] bit is asserted the first (lowest number) active Tx Mailbox found is the arbitration winner. MCR[LPRIOEN] bit has no effect when CTRL1[LBUF] is asserted.

### 37.5.2.2 Highest-priority Mailbox first

If CTRL1[LBUF] bit is negated, then the arbitration process searches the active Tx Mailbox with the highest priority, which means that this Mailbox's frame would have a higher probability to win the arbitration on CAN bus when multiple external nodes compete for the bus at the same time.

The sequence of bits considered for this arbitration is called the *arbitration value* of the Mailbox. The highest-priority Tx Mailbox is the one that has the lowest arbitration value among all Tx Mailboxes.

If two or more Mailboxes have equivalent arbitration values, the Mailbox with the lowest number is the arbitration winner.

The composition of the arbitration value depends on MCR[LPRIOEN] bit setting.

#### 37.5.2.2.1 Local Priority disabled

If MCR[LPRIOEN] bit is negated the arbitration value is built in the exact sequence of bits as they would be transmitted in a CAN frame (see the following table) in such a way that the Local Priority is disabled.

**Table 37-102. Composition of the arbitration value when Local Priority is disabled**

| Format             | Mailbox Arbitration Value (32 bits) |             |             |                             |             |
|--------------------|-------------------------------------|-------------|-------------|-----------------------------|-------------|
| Standard (IDE = 0) | Standard ID (11 bits)               | RTR (1 bit) | IDE (1 bit) | - (18 bits)                 | - (1 bit)   |
| Extended (IDE = 1) | Extended ID[28:18] (11 bits)        | SRR (1 bit) | IDE (1 bit) | Extended ID[17:0] (18 bits) | RTR (1 bit) |

#### 37.5.2.2.2 Local Priority enabled

If Local Priority is desired MCR[LPRIOEN] must be asserted. In this case the Mailbox PRIO field is included at the very left of the arbitration value (see the following table).

**Table 37-103. Composition of the arbitration value when Local Priority is enabled**

| Format             | Mailbox Arbitration Value (35 bits) |                              |             |             |                             |             |
|--------------------|-------------------------------------|------------------------------|-------------|-------------|-----------------------------|-------------|
| Standard (IDE = 0) | PRIO (3 bits)                       | Standard ID (11 bits)        | RTR (1 bit) | IDE (1 bit) | - (18 bits)                 | - (1 bit)   |
| Extended (IDE = 1) | PRIO (3 bits)                       | Extended ID[28:18] (11 bits) | SRR (1 bit) | IDE (1 bit) | Extended ID[17:0] (18 bits) | RTR (1 bit) |

As the PRIO field is the most significant part of the arbitration value Mailboxes with low PRIO values have higher priority than Mailboxes with high PRIO values regardless the rest of their arbitration values.

Note that the PRIO field is not part of the frame on the CAN bus. Its purpose is only to affect the internal arbitration process.

### 37.5.2.3 Arbitration process (continued)

After the arbitration winner is found, its content is copied to a hidden auxiliary MB called Tx Serial Message Buffer (Tx SMB), which has the same structure as a normal MB but is not user accessible. This operation is called move-out and after it is done, write access to the corresponding MB is blocked (if the AEN bit in MCR is asserted). The write access is released in the following events:

- After the MB is transmitted
- FlexCAN enters in Freeze mode or Bus Off
- FlexCAN loses the bus arbitration or there is an error during the transmission

At the first opportunity window on the CAN bus, the message on the Tx SMB is transmitted according to the CAN protocol rules. FlexCAN transmits up to eight data bytes, even if the Data Length Code (DLC) field value is greater than that.

Arbitration process can be triggered in the following situations:

- During Rx and Tx frames from CAN CRC field to end of frame. T ASD value may be changed to optimize the arbitration start point.
- During CAN BusOff state from TX\_ERR\_CNT=124 to 128. T ASD value may be changed to optimize the arbitration start point.
- During C/S write by CPU in BusIdle. First C/S write starts arbitration process and a second C/S write during this same arbitration restarts the process. If other C/S writes are performed, Tx arbitration process is pending. If there is no arbitration winner after arbitration process has finished, then TX arbitration machine begins a new arbitration process.
  - If there is a pending arbitration and BusIdle state starts then an arbitration process is triggered. In this case the first and second C/S write in BusIdle will not restart the arbitration process. It is possible that there is not enough time to finish arbitration in WaitForBusIdle state and the next state is Idle. In this case the scan is not interrupted, and it is completed during BusIdle state. During this arbitration C/S write does not cause arbitration restart.
- Arbitration winner deactivation during a valid arbitration window.
- Upon Leave Freeze mode (first bit of the WaitForBusIdle state). If there is a re-synchronization during WaitForBusIdle arbitration process is restarted.

Arbitration process stops in the following situation:

- All Mailboxes were scanned
- A Tx active Mailbox is found in case of Lowest Buffer feature enabled
- Arbitration winner inactivation or abort during any arbitration process
- There was not enough time to finish Tx arbitration process (for instance, when a deactivation was performed near the end of frame). In this case arbitration process is pending.
- Error or Overload flag in the bus
- Low Power or Freeze mode request in Idle state

Arbitration is considered pending as described below:

- It was not possible to finish arbitration process in time
- C/S write during arbitration if write is performed in a MB whose number is lower than the Tx arbitration pointer
- Any C/S write if there is no Tx Arbitration process in progress
- Rx Match has just updated a Rx Code to Tx Code
- Entering Busoff state

C/S write during arbitration has the following effect:

- If C/S write is performed in the arbitration winner, a new process is restarted immediately.
- If C/S write is performed in a MB whose number is higher than the Tx arbitration pointer, the ongoing arbitration process will scan this MB as normal.

### 37.5.3 Receive process

To be able to receive CAN frames into a Mailbox, the CPU must prepare it for reception by executing the following steps:

1. If the Mailbox is active (either Tx or Rx) inactivate the Mailbox (see [Mailbox inactivation](#)), preferably with a safe inactivation (see [Transmission abort mechanism](#)).
2. Write the ID word
3. Write the EMPTY code (0b0100) to the CODE field of the Control and Status word to activate the Mailbox.

After the MB is activated, it will be able to receive frames that match the programmed filter. At the end of a successful reception, the Mailbox is updated by the *move-in* process (see [Move-in](#)) as follows:

1. The received Data field (8 bytes at most) is stored.
2. The received Identifier field is stored.
3. The value of the Free Running Timer at the time of the second bit of frame's Identifier field is written into the Mailbox's Time Stamp field.
4. The received SRR, IDE, RTR, and DLC fields are stored.
5. The CODE field in the Control and Status word is updated (see [Table 37-97](#) and [Table 37-98](#) in Section [Message buffer structure](#)).
6. A status flag is set in the Interrupt Flag Register and an interrupt is generated if allowed by the corresponding Interrupt Mask Register bit.

The recommended way for CPU servicing (read) the frame received in an Mailbox is using the following procedure:

1. Read the Control and Status word of that Mailbox.
2. Check if the BUSY bit is deasserted, indicating that the Mailbox is locked. Repeat step 1) while it is asserted. See [Mailbox lock mechanism](#).
3. Read the contents of the Mailbox. Once Mailbox is locked now, its contents won't be modified by FlexCAN Move-in processes. See [Move-in](#).
4. Acknowledge the proper flag at IFLAG registers.
5. Read the Free Running Timer. It is optional but recommended to unlock Mailbox as soon as possible and make it available for reception.

The CPU should synchronize to frame reception by the status flag bit for the specific Mailbox in one of the IFLAG Registers and not by the CODE field of that Mailbox. Polling the CODE field does not work because once a frame was received and the CPU services the Mailbox (by reading the C/S word followed by unlocking the Mailbox), the CODE field will not return to EMPTY. It will remain FULL, as explained in [Table 37-97](#). If the CPU tries to workaround this behavior by writing to the C/S word to force an EMPTY code after reading the Mailbox without a prior *safe inactivation*, a newly received frame matching the filter of that Mailbox may be lost.

### CAUTION

*In summary: never do polling by reading directly the C/S word of the Mailboxes. Instead, read the IFLAG registers.*

Note that the received frame's Identifier field is always stored in the matching Mailbox, thus the contents of the ID field in an Mailbox may change if the match was due to masking. Note also that FlexCAN does receive frames transmitted by itself if there exists a matching Rx Mailbox, provided the MCR[SRXDIS] bit is not asserted. If the MCR[SRXDIS] bit is asserted, FlexCAN will not store frames transmitted by itself in any MB, even if it contains a matching MB, and no interrupt flag or interrupt signal will be generated due to the frame reception.

To be able to receive CAN frames through the Rx FIFO, the CPU must enable and configure the Rx FIFO during Freeze mode (see [Rx FIFO](#)). Upon receiving the Frames Available in Rx FIFO interrupt (see the description of the IFLAG[BUF5I] "Frames available in Rx FIFO" bit in the IMASK1 register), the CPU should service the received frame using the following procedure:

1. Read the Control and Status word (optional – needed only if a mask was used for IDE and RTR bits)
2. Read the ID field (optional – needed only if a mask was used)
3. Read the Data field
4. Read the RXFIR register (optional)
5. Clear the Frames Available in Rx FIFO interrupt by writing 1 to IFLAG[BUF5I] bit (mandatory – releases the MB and allows the CPU to read the next Rx FIFO entry)

### 37.5.4 Matching process

The matching process scans the MB memory looking for Rx MBs programmed with the same ID as the one received from the CAN bus. If the FIFO is enabled, the priority of scanning can be selected between Mailboxes and FIFO filters. In any case, the matching starts from the lowest number Message Buffer toward the higher ones. If no match is found within the first structure then the other is scanned subsequently. In the event that the FIFO is full, the matching algorithm will always look for a matching MB outside the FIFO region.

As the frame is being received, it is stored in a hidden auxiliary MB called Rx Serial Message Buffer (Rx SMB).

The matching process start point depends on the following conditions:

- If the received frame is a remote frame, the start point is the CRC field of the frame



- If the received frame is a data frame with DLC field equal to zero, the start point is the CRC field of the frame
- If the received frame is a data frame with DLC field different than zero, the start point is the DATA field of the frame

If a matching ID is found in the FIFO table or in one of the Mailboxes, the contents of the SMB will be transferred to the FIFO or to the matched Mailbox by the move-in process. If any CAN protocol error is detected then no match results will be transferred to the FIFO or to the matched Mailbox at the end of reception.

The matching process scans all matching elements of both Rx FIFO (if enabled) and active Rx Mailboxes (CODE is EMPTY, FULL, OVERRUN or RANSWER) in search of a successful comparison with the matching elements of the Rx SMB that is receiving the frame on the CAN bus. The SMB has the same structure of a Mailbox. The reception structures (Rx FIFO or Mailboxes) associated with the matching elements that had a successful comparison are the *matched structures*. The *matching winner* is selected at the end of the scan among those matched structures and depends on conditions described ahead. See the following table.

**Table 37-104. Matching architecture**

| Structure         | SMB[RTR] | CTRL2[RRS] | CTRL2[EAC EN] | MB[IDE]          | MB[RTR]             | MB[ID <sup>1</sup> ] | MB[CODE]                 |
|-------------------|----------|------------|---------------|------------------|---------------------|----------------------|--------------------------|
| Mailbox           | 0        | -          | 0             | cmp <sup>2</sup> | no_cmp <sup>3</sup> | cmp_msk <sup>4</sup> | EMPTY or FULL or OVERRUN |
| Mailbox           | 0        | -          | 1             | cmp_msk          | cmp_msk             | cmp_msk              | EMPTY or FULL or OVERRUN |
| Mailbox           | 1        | 0          | -             | cmp              | no_cmp              | cmp                  | RANSWER                  |
| Mailbox           | 1        | 1          | 0             | cmp              | no_cmp              | cmp_msk              | EMPTY or FULL or OVERRUN |
| Mailbox           | 1        | 1          | 1             | cmp_msk          | cmp_msk             | cmp_msk              | EMPTY or FULL or OVERRUN |
| FIFO <sup>5</sup> | -        | -          | -             | cmp_msk          | cmp_msk             | cmp_msk              | -                        |

1. For Mailbox structure, If SMB[IDE] is asserted, the ID is 29 bits (ID Standard + ID Extended). If SMB[IDE] is negated, the ID is only 11 bits (ID Standard). For FIFO structure, the ID depends on IDAM.
2. cmp: Compares the SMB contents with the MB contents regardless the masks.
3. no\_cmp: The SMB contents are not compared with the MB contents
4. cmp\_msk: Compares the SMB contents with MB contents taking into account the masks.
5. SMB[IDE] and SMB[RTR] are not taken into account when IDAM is type C.

A reception structure is *free-to-receive* when any of the following conditions is satisfied:

- The CODE field of the Mailbox is EMPTY

- The CODE field of the Mailbox is either FULL or OVERRUN and it has already been serviced (the C/S word was read by the CPU and unlocked as described in [Mailbox lock mechanism](#))
- The CODE field of the Mailbox is either FULL or OVERRUN and an inactivation (see [Mailbox inactivation](#)) is performed
- The Rx FIFO is not full

The scan order for Mailboxes and Rx FIFO is from the matching element with lowest number to the higher ones.

The matching winner search for Mailboxes is affected by the MCR[IRMQ] bit. If it is negated the matching winner is the first matched Mailbox regardless if it is free-to-receive or not. If it is asserted, the matching winner is selected according to the priority below:

1. the first free-to-receive matched Mailbox;
2. the last non free-to-receive matched Mailbox.

It is possible to select the priority of scan between Mailboxes and Rx FIFO by the CTRL2[MRP] bit.

If the selected priority is Rx FIFO first:

- If the Rx FIFO is a matched structure and is free-to-receive then the Rx FIFO is the matching winner regardless of the scan for Mailboxes
- Otherwise (the Rx FIFO is not a matched structure or is not free-to-receive), then the matching winner is searched among Mailboxes as described above

If the selected priority is Mailboxes first:

- If a free-to-receive matched Mailbox is found, it is the matching winner regardless the scan for Rx FIFO
- If no matched Mailbox is found, then the matching winner is searched in the scan for the Rx FIFO
- If both conditions above are not satisfied and a non free-to-receive matched Mailbox is found then the matching winner determination is conditioned by the MCR[IRMQ] bit:
  - If MCR[IRMQ] bit is negated the matching winner is the first matched Mailbox
  - If MCR[IRMQ] bit is asserted the matching winner is the Rx FIFO if it is a free-to-receive matched structure, otherwise the matching winner is the last non free-to-receive matched Mailbox

See the following table for a summary of matching possibilities.

**Table 37-105. Matching possibilities and resulting reception structures**

| RFEN                                                               | IRMQ | MRP            | Matched in MB     | Matched in FIFO      | Reception structure | Description                             |
|--------------------------------------------------------------------|------|----------------|-------------------|----------------------|---------------------|-----------------------------------------|
| <b>No FIFO, only MB, match is always MB first</b>                  |      |                |                   |                      |                     |                                         |
| 0                                                                  | 0    | X <sup>1</sup> | None <sup>2</sup> | - <sup>3</sup>       | None                | Frame lost by no match                  |
| 0                                                                  | 0    | X              | Free <sup>4</sup> | -                    | FirstMB             |                                         |
| 0                                                                  | 1    | X              | None              | -                    | None                | Frame lost by no match                  |
| 0                                                                  | 1    | X              | Free              | -                    | FirstMb             |                                         |
| 0                                                                  | 1    | X              | NotFree           | -                    | LastMB              | Overrun                                 |
| <b>FIFO enabled, no match in FIFO is as if FIFO does not exist</b> |      |                |                   |                      |                     |                                         |
| 1                                                                  | 0    | X              | None              | None <sup>5</sup>    | None                | Frame lost by no match                  |
| 1                                                                  | 0    | X              | Free              | None                 | FirstMB             |                                         |
| 1                                                                  | 1    | X              | None              | None                 | None                | Frame lost by no match                  |
| 1                                                                  | 1    | X              | Free              | None                 | FirstMb             |                                         |
| 1                                                                  | 1    | X              | NotFree           | None                 | LastMB              | Overrun                                 |
| <b>FIFO enabled, Queue disabled</b>                                |      |                |                   |                      |                     |                                         |
| 1                                                                  | 0    | 0              | X                 | NotFull <sup>6</sup> | FIFO                |                                         |
| 1                                                                  | 0    | 0              | None              | Full <sup>7</sup>    | None                | Frame lost by FIFO full (FIFO Overflow) |
| 1                                                                  | 0    | 0              | Free              | Full                 | FirstMB             |                                         |
| 1                                                                  | 0    | 0              | NotFree           | Full                 | FirstMB             |                                         |
| 1                                                                  | 0    | 1              | None              | NotFull              | FIFO                |                                         |
| 1                                                                  | 0    | 1              | None              | Full                 | None                | Frame lost by FIFO full (FIFO Overflow) |
| 1                                                                  | 0    | 1              | Free              | X                    | FirstMB             |                                         |
| 1                                                                  | 0    | 1              | NotFree           | X                    | FirstMb             | Overrun                                 |
| <b>FIFO enabled, Queue enabled</b>                                 |      |                |                   |                      |                     |                                         |
| 1                                                                  | 1    | 0              | X                 | NotFull              | FIFO                |                                         |
| 1                                                                  | 1    | 0              | None              | Full                 | None                | Frame lost by FIFO full (FIFO Overflow) |
| 1                                                                  | 1    | 0              | Free              | Full                 | FirstMB             |                                         |
| 1                                                                  | 1    | 0              | NotFree           | Full                 | LastMb              | Overrun                                 |
| 1                                                                  | 1    | 1              | None              | NotFull              | FIFO                |                                         |
| 1                                                                  | 1    | 1              | Free              | X                    | FirstMB             |                                         |
| 1                                                                  | 1    | 1              | NotFree           | NotFull              | FIFO                |                                         |
| 1                                                                  | 1    | 1              | NotFree           | Full                 | LastMb              | Overrun                                 |

1. This is a don't care condition.

2. Matched in MB "None" means that the frame has not matched any MB (free-to-receive or non-free-to-receive).

## Functional description

3. This is a forbidden condition.
4. Matched in MB "Free" means that the frame matched at least one MB free-to-receive regardless of whether it has matched MBs non-free-to-receive.
5. Matched in FIFO "None" means that the frame has not matched any filter in FIFO. It is as if the FIFO didn't exist (CTRL2[RFEN]=0).
6. Matched in FIFO "NotFull" means that the frame has matched a FIFO filter and has empty slots to receive it.
7. Matched in FIFO "Full" means that the frame has matched a FIFO filter but couldn't store it because it has no empty slots to receive it.

If a non-safe Mailbox inactivation (see [Mailbox inactivation](#)) occurs during matching process and the Mailbox inactivated is the temporary matching winner then the temporary matching winner is invalidated. The matching elements scan is not stopped nor restarted, it continues normally. The consequence is that the current matching process works as if the matching elements compared before the inactivation did not exist, therefore a message may be lost.

Suppose, for example, that the FIFO is disabled, IRMQ is enabled and there are two MBs with the same ID, and FlexCAN starts receiving messages with that ID. Let us say that these MBs are the second and the fifth in the array. When the first message arrives, the matching algorithm will find the first match in MB number 2. The code of this MB is EMPTY, so the message is stored there. When the second message arrives, the matching algorithm will find MB number 2 again, but it is not "free-to-receive", so it will keep looking and find MB number 5 and store the message there. If yet another message with the same ID arrives, the matching algorithm finds out that there are no matching MBs that are "free-to-receive", so it decides to overwrite the last matched MB, which is number 5. In doing so, it sets the CODE field of the MB to indicate OVERRUN.

The ability to match the same ID in more than one MB can be exploited to implement a reception queue (in addition to the full featured FIFO) to allow more time for the CPU to service the MBs. By programming more than one MB with the same ID, received messages will be queued into the MBs. The CPU can examine the Time Stamp field of the MBs to determine the order in which the messages arrived.

Matching to a range of IDs is possible by using ID Acceptance Masks. FlexCAN supports individual masking per MB. Refer to the description of the Rx Individual Mask Registers (RXIMRx). During the matching algorithm, if a mask bit is asserted, then the corresponding ID bit is compared. If the mask bit is negated, the corresponding ID bit is "don't care". Please note that the Individual Mask Registers are implemented in RAM, so they are not initialized out of reset. Also, they can only be programmed while the module is in Freeze mode; otherwise, they are blocked by hardware.

FlexCAN also supports an alternate masking scheme with only four mask registers (RXGMASK, RX14MASK, RX15MASK and RXFGMASK) for backwards compatibility with legacy applications. This alternate masking scheme is enabled when the IRMQ bit in the MCR Register is negated.

### 37.5.5 Move process

There are two types of move process: move-in and move-out.

#### 37.5.5.1 Move-in

The move-in process is the copy of a message received by an Rx SMB to a Rx Mailbox or FIFO that has matched it. If the move destination is the Rx FIFO, attributes of the message are also copied to the RXFIR FIFO. Each Rx SMB has its own move-in process, but only one is performed at a given time as described ahead. The move-in starts only when the message held by the Rx SMB has a corresponding matching winner (see [Matching process](#)) and all of the following conditions are true:

- The CAN bus has reached or let past either:
  - The second bit of Intermission field next to the frame that carried the message that is in the Rx SMB
  - The first bit of an overload frame next to the frame that carried the message that is in the Rx SMB
- There is no ongoing matching process
- The destination Mailbox is not locked by the CPU
- There is no ongoing move-in process from another Rx SMB. If more than one move-in processes are to be started at the same time both are performed and the newest substitutes the oldest.

The term *pending move-in* is used throughout the documentation and stands for a move-to-be that still does not satisfy all of the aforementioned conditions.

The move-in is cancelled and the Rx SMB is able to receive another message if any of the following conditions is satisfied:

- The destination Mailbox is inactivated after the CAN bus has reached the first bit of Intermission field next to the frame that carried the message and its matching process has finished
- There is a previous pending move-in to the same destination Mailbox
- The Rx SMB is receiving a frame transmitted by the FlexCAN itself and the self-reception is disabled (MCR[SRXDIS] bit is asserted)
- Any CAN protocol error is detected

Note that the pending move-in is not cancelled if the module enters Freeze or Low-Power mode. It only stays on hold waiting for exiting Freeze and Low-Power mode and to be unlocked. If an MB is unlocked during Freeze mode, the move-in happens immediately.

The move-in process is the execution by the FlexCAN of the following steps:

1. if the message is destined to the Rx FIFO, push IDHIT into the RXFIR FIFO;
2. reads the words DATA0-3 and DATA4-7 from the Rx SMB;
3. writes it in the words DATA0-3 and DATA4-7 of the Rx Mailbox;
4. reads the words Control/Status and ID from the Rx SMB;
5. writes it in the words Control/Status and ID of the Rx Mailbox, updating the CODE field.

The move-in process is not atomic, in such a way that it is immediately cancelled by the inactivation of the destination Mailbox (see [Mailbox inactivation](#)) and in this case the Mailbox may be left partially updated, thus incoherent. The exception is if the move-in destination is an Rx FIFO Message Buffer, then the process cannot be cancelled.

The BUSY Bit (least significant bit of the CODE field) of the destination Message Buffer is asserted while the move-in is being performed to alert the CPU that the Message Buffer content is temporarily incoherent.

### **37.5.5.2 Move-out**

The move-out process is the copy of the content from a Tx Mailbox to the Tx SMB when a message for transmission is available (see Section "Arbitration process"). The move-out occurs in the following conditions:

- The first bit of Intermission field
- During Bus Off state when TX Error Counter is in the 124 to 128 range
- During Bus Idle state
- During Wait For Bus Idle state

The move-out process is not atomic. Only the CPU has priority to access the memory concurrently out of Bus Idle state. In Bus Idle, the move-out has the lowest priority to the concurrent memory accesses.

### **37.5.6 Data coherence**

In order to maintain data coherency and FlexCAN proper operation, the CPU must obey the rules described in [Transmit process](#) and [Receive process](#).

### 37.5.6.1 Transmission abort mechanism

The abort mechanism provides a safe way to request the abortion of a pending transmission. A feedback mechanism is provided to inform the CPU if the transmission was aborted or if the frame could not be aborted and was transmitted instead.

Two primary conditions must be fulfilled in order to abort a transmission:

- MCR[AEN] bit must be asserted
- The first CPU action must be the writing of abort code (0b1001) into the CODE field of the Control and Status word.

The active MBs configured as transmission must be aborted first and then they may be updated. If the abort code is written to a Mailbox that is currently being transmitted, or to a Mailbox that was already loaded into the SMB for transmission, the write operation is blocked and the MB is kept active, but the abort request is captured and kept pending until one of the following conditions are satisfied:

- The module loses the bus arbitration
- There is an error during the transmission
- The module is put into Freeze mode
- The module enters in BusOff state
- There is an overload frame

If none of the conditions above are reached, the MB is transmitted correctly, the interrupt flag is set in the IFLAG register, and an interrupt to the CPU is generated (if enabled). The abort request is automatically cleared when the interrupt flag is set. On the other hand, if one of the above conditions is reached, the frame is not transmitted; therefore, the abort code is written into the CODE field, the interrupt flag is set in the IFLAG, and an interrupt is (optionally) generated to the CPU.

If the CPU writes the abort code before the transmission begins internally, then the write operation is not blocked; therefore, the MB is updated and the interrupt flag is set. In this way the CPU just needs to read the abort code to make sure the active MB was *safely inactivated*. Although the AEN bit is asserted and the CPU wrote the abort code, in this case the MB is inactivated and not aborted, because the transmission did not start yet. One Mailbox is only aborted when the abort request is captured and kept pending until one of the previous conditions are satisfied.

The abort procedure can be summarized as follows:

- CPU checks the corresponding IFLAG and clears it, if asserted.

- CPU writes 0b1001 into the CODE field of the C/S word.
- CPU waits for the corresponding IFLAG indicating that the frame was either transmitted or aborted.
- CPU reads the CODE field to check if the frame was either transmitted (CODE=0b1000) or aborted (CODE=0b1001).
- It is necessary to clear the corresponding IFLAG in order to allow the MB to be reconfigured.

### 37.5.6.2 Mailbox inactivation

Inactivation is a mechanism provided to protect the Mailbox against updates by the FlexCAN internal processes, thus allowing the CPU to rely on Mailbox data coherence after having updated it, even in Normal mode.

Inactivation of transmission Mailboxes must be performed just when MCR[AEN] bit is deasserted.

If a Mailbox is inactivated, it participates in neither the arbitration process nor the matching process until it is reactivated. See [Transmit process](#) and [Receive process](#) for more detailed instruction on how to inactivate and reactivate a Mailbox.

To inactivate a Mailbox, the CPU must update its CODE field to INACTIVE (either 0b0000 or 0b1000).

Because the user is not able to synchronize the CODE field update with the FlexCAN internal processes, an inactivation can lead to undesirable results:

- A frame in the bus that matches the filtering of the inactivated Rx Mailbox may be lost without notice, even if there are other Mailboxes with the same filter
- A frame containing the message within the inactivated Tx Mailbox may be transmitted without notice

In order to eliminate such risk and perform a *safe inactivation* the CPU must use the following mechanism along with the inactivation itself:

- For Tx Mailboxes, the Transmission Abort (see [Transmission abort mechanism](#))

The inactivation automatically unlocks the Mailbox (see [Mailbox lock mechanism](#)).

#### NOTE

Message Buffers that are part of the Rx FIFO cannot be inactivated. There is no write protection on the FIFO region by



FlexCAN. CPU must maintain data coherency in the FIFO region when RFEN is asserted.

### 37.5.6.3 Mailbox lock mechanism

Other than Mailbox inactivation, FlexCAN has another data coherence mechanism for the receive process. When the CPU reads the Control and Status word of an Rx MB with codes FULL or OVERRUN, FlexCAN assumes that the CPU wants to read the whole MB in an atomic operation, and therefore it sets an internal lock flag for that MB. The lock is released when the CPU reads the Free Running Timer (global unlock operation), or when it reads the Control and Status word of another MB regardless of its code. A CPU write into C/S word also unlocks the MB, but this procedure is not recommended for normal unlock use because it cancels a pending-move and potentially may lose a received message. The MB locking is done to prevent a new frame to be written into the MB while the CPU is reading it.

#### NOTE

The locking mechanism applies only to Rx MBs that are not part of FIFO and have a code different than INACTIVE (0b0000) or EMPTY<sup>1</sup> (0b0100). Also, Tx MBs can not be locked.

Suppose, for example, that the FIFO is disabled and the second and the fifth MBs of the array are programmed with the same ID, and FlexCAN has already received and stored messages into these two MBs. Suppose now that the CPU decides to read MB number 5 and at the same time another message with the same ID is arriving. When the CPU reads the Control and Status word of MB number 5, this MB is locked. The new message arrives and the matching algorithm finds out that there are no "free-to-receive" MBs, so it decides to override MB number 5. However, this MB is locked, so the new message can not be written there. It will remain in the SMB waiting for the MB to be unlocked, and only then will be written to the MB.

If the MB is not unlocked in time and yet another new message with the same ID arrives, then the new message overwrites the one on the SMB and there will be no indication of lost messages either in the CODE field of the MB or in the Error and Status Register.

While the message is being moved-in from the SMB to the MB, the BUSY bit on the CODE field is asserted. If the CPU reads the Control and Status word and finds out that the BUSY bit is set, it should defer accessing the MB until the BUSY bit is negated.

---

1. In previous FlexCAN versions, reading the C/S word locked the MB even if it was EMPTY. This behavior is maintained when the IRMQ bit is negated.

### Note

If the BUSY bit is asserted or if the MB is empty, then reading the Control and Status word does not lock the MB.

Inactivation takes precedence over locking. If the CPU inactivates a locked Rx MB, then its lock status is negated and the MB is marked as invalid for the current matching round. Any pending message on the SMB will not be transferred anymore to the MB. An MB is unlocked when the CPU reads the Free Running Timer Register (see Section "Free Running Timer Register (TIMER)"), or the C/S word of another MB.

Lock and unlock mechanisms have the same functionality in both Normal and Freeze modes.

An unlock during Normal or Freeze mode results in the move-in of the pending message. However, the move-in is postponed if an unlock occurs during a low power mode (see [Modes of operation](#)) and it will take place only when the module resumes to Normal or Freeze modes.

## 37.5.7 Rx FIFO

The Rx FIFO is receive-only and is enabled by asserting the MCR[RFEN] bit. The reset value of this bit is zero to maintain software backward compatibility with previous versions of the module that did not have the FIFO feature. The FIFO is 6-message deep. The memory region occupied by the FIFO structure (both Message Buffers and FIFO engine) is described in [Rx FIFO structure](#). The CPU can read the received messages sequentially, in the order they were received, by repeatedly reading a Message Buffer structure at the output of the FIFO.

The IFLAG[BUF5I] (Frames available in Rx FIFO) is asserted when there is at least one frame available to be read from the FIFO. An interrupt is generated if it is enabled by the corresponding mask bit. Upon receiving the interrupt, the CPU can read the message (accessing the output of the FIFO as a Message Buffer) and the RXFIR register and then clear the interrupt. If there are more messages in the FIFO the act of clearing the interrupt updates the output of the FIFO with the next message and update the RXFIR with the attributes of that message, reissuing the interrupt to the CPU. Otherwise, the flag remains negated. The output of the FIFO is only valid whilst the IFLAG[BUF5I] is asserted.

The IFLAG[BUF6I] (Rx FIFO Warning) is asserted when the number of unread messages within the Rx FIFO is increased to 5 from 4 due to the reception of a new one, meaning that the Rx FIFO is almost full. The flag remains asserted until the CPU clears it.

The IFLAG[BUF7I] (Rx FIFO Overflow) is asserted when an incoming message was lost because the Rx FIFO is full. Note that the flag will not be asserted when the Rx FIFO is full and the message was captured by a Mailbox. The flag remains asserted until the CPU clears it.

Clearing one of those three flags does not affect the state of the other two.

An interrupt is generated if an IFLAG bit is asserted and the corresponding mask bit is asserted too.

A powerful filtering scheme is provided to accept only frames intended for the target application, reducing the interrupt servicing work load. The filtering criteria is specified by programming a table of up to 128 32-bit registers, according to CTRL2[RFFN] setting, that can be configured to one of the following formats (see also [Rx FIFO structure](#)):

- Format A: 128 IDAFs (extended or standard IDs including IDE and RTR)
- Format B: 256 IDAFs (standard IDs or extended 14-bit ID slices including IDE and RTR)
- Format C: 512 IDAFs (standard or extended 8-bit ID slices)

### Note

A chosen format is applied to all entries of the filter table. It is not possible to mix formats within the table.

Every frame available in the FIFO has a corresponding IDHIT (Identifier Acceptance Filter Hit Indicator) that can be read by accessing the RXFIR register. The RXFIR[IDHIT] field refers to the message at the output of the FIFO and is valid while the IFLAG[BUF5I] flag is asserted. The RXFIR register must be read only before clearing the flag, which guarantees that the information refers to the correct frame within the FIFO.

Up to 32 elements of the filter table are individually affected by the Individual Mask Registers (RXIMRx), according to the setting of CTRL2[RFFN], allowing very powerful filtering criteria to be defined. If the IRMQ bit is negated, then the FIFO filter table is affected by RXFGMASK.

## 37.5.8 CAN protocol related features

This section describes the CAN protocol related features.

### 37.5.8.1 Remote frames

Remote frame is a special kind of frame. The user can program a mailbox to be a Remote Request Frame by writing the mailbox as Transmit with the RTR bit set to '1'. After the remote request frame is transmitted successfully, the mailbox becomes a Receive Message Buffer, with the same ID as before.

When a remote request frame is received by FlexCAN, it can be treated in three ways, depending on Remote Request Storing (CTRL2[RRS]) and Rx FIFO Enable (MCR[RFEN]) bits:

- If RRS is negated the frame's ID is compared to the IDs of the Transmit Message Buffers with the CODE field 0b1010. If there is a matching ID, then this mailbox frame will be transmitted. Note that if the matching mailbox has the RTR bit set, then FlexCAN will transmit a remote frame as a response. The received remote request frame is not stored in a receive buffer. It is only used to trigger a transmission of a frame in response. The mask registers are not used in remote frame matching, and all ID bits (except RTR) of the incoming received frame should match. In the case that a remote request frame was received and matched a mailbox, this message buffer immediately enters the internal arbitration process, but is considered as normal Tx mailbox, with no higher priority. The data length of this frame is independent of the DLC field in the remote frame that initiated its transmission.
- If RRS is asserted the frame's ID is compared to the IDs of the receive mailboxes with the CODE field 0b0100, 0b0010 or 0b0110. If there is a matching ID, then this mailbox will store the remote frame in the same fashion of a data frame. No automatic remote response frame will be generated. The mask registers are used in the matching process.
- If RFEN is asserted FlexCAN will not generate an automatic response for remote request frames that match the FIFO filtering criteria. If the remote frame matches one of the target IDs, it will be stored in the FIFO and presented to the CPU. Note that for filtering formats A and B, it is possible to select whether remote frames are accepted or not. For format C, remote frames are always accepted (if they match the ID). Remote Request Frames are considered as normal frames, and generate a FIFO overflow when a successful reception occurs and the FIFO is already full.

### 37.5.8.2 Overload frames

FlexCAN does transmit overload frames due to detection of following conditions on CAN bus:

- Detection of a dominant bit in the first/second bit of Intermission
- Detection of a dominant bit at the 7th bit (last) of End of Frame field (Rx frames)
- Detection of a dominant bit at the 8th bit (last) of Error Frame Delimiter or Overload Frame Delimiter

### 37.5.8.3 Time stamp

The value of the Free Running Timer is sampled at the beginning of the Identifier field on the CAN bus, and is stored at the end of "move-in" in the TIME STAMP field, providing network behavior with respect to time.

The Free Running Timer can be reset upon a specific frame reception, enabling network time synchronization. See the TSYN description in the description of the Control 1 Register (CTRL1).

### 37.5.8.4 Protocol timing

The following figure shows the structure of the clock generation circuitry that feeds the CAN Protocol Engine (PE) submodule. The clock source bit CLKSRC in the CTRL1 Register defines whether the internal clock is connected to the output of a crystal oscillator (Oscillator Clock) or to the Peripheral Clock. In order to guarantee reliable operation, the clock source should be selected while the module is in Disable Mode (bit MDIS set in the Module Configuration Register).

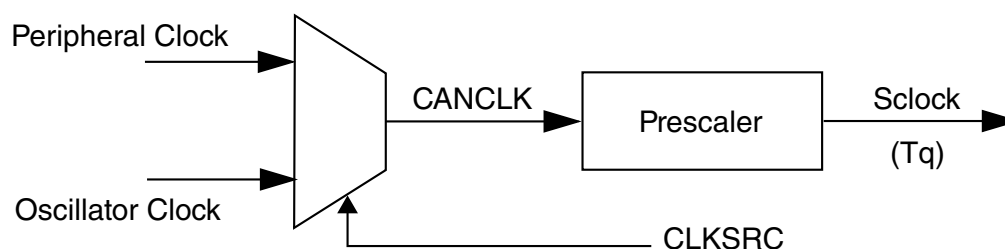


**Figure 37-92. CAN engine clocking scheme**

The oscillator clock should be selected whenever a tight tolerance (up to 0.1%) is required in the CAN bus timing. The oscillator clock has better jitter performance.

The FlexCAN module supports a variety of means to setup bit timing parameters that are required by the CAN protocol. The Control Register has various fields used to control bit timing parameters: PRESDIV, PROPSEG, PSEG1, PSEG2 and RJW. See the description of the Control 1 Register (CTRL1).

The PRESDIV field controls a prescaler that generates the Serial Clock (Sclock), whose period defines the 'time quantum' used to compose the CAN waveform. A time quantum is the atomic unit of time handled by the CAN engine.

$$f_{Tq} = \frac{f_{CANCLK}}{(\text{Prescaler Value})}$$

A bit time is subdivided into three segments<sup>1</sup> (see [Figure 37-93](#) and [Table 37-106](#)):

- SYNC\_SEG: This segment has a fixed length of one time quantum. Signal edges are expected to happen within this section
- Time Segment 1: This segment includes the Propagation Segment and the Phase Segment 1 of the CAN standard. It can be programmed by setting the PROPSEG and the PSEG1 fields of the CTRL1 Register so that their sum (plus 2) is in the range of 4 to 16 time quanta
- Time Segment 2: This segment represents the Phase Segment 2 of the CAN standard. It can be programmed by setting the PSEG2 field of the CTRL1 Register (plus 1) to be 2 to 8 time quanta long

$$\text{Bit Rate} = \frac{f_{Tq}}{(\text{number of Time Quanta})}$$

1. For further explanation of the underlying concepts, see ISO/DIS 11519–1, Section 10.3. See also the CAN 2.0A/B protocol specification for bit timing.

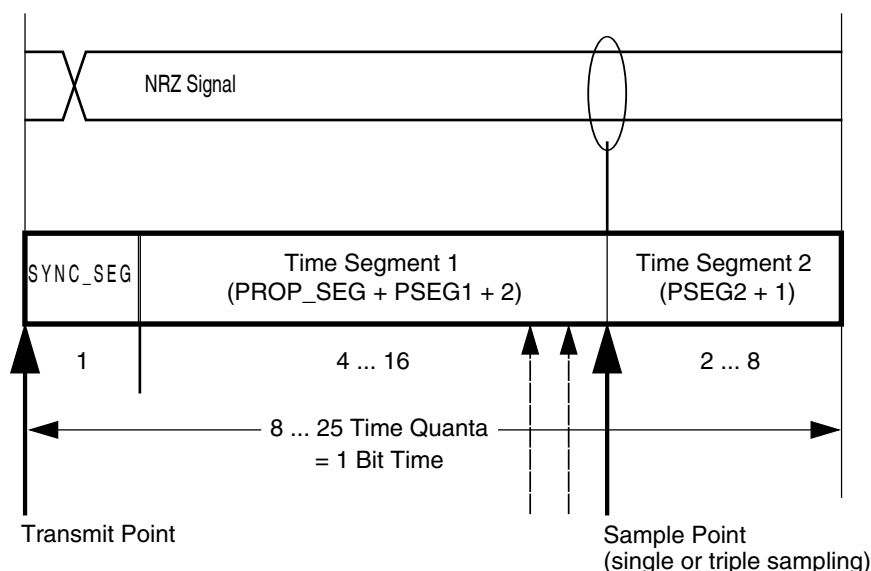


**Figure 37-93. Segments within the bit time**

Whenever CAN bit is used as a measure of duration (for example, MCR[FRZACK] and MCR[LPMACK]), the number of peripheral clocks in one CAN bit can be calculated as:

$$NCCP = \frac{f_{SYS} \times [1 + (PSEG1 + 1) + (PSEG2 + 1) + (PROPSEG + 1)] \times (PRES DIV + 1)}{f_{CANCLK}}$$

where:

- NCCP is the number of peripheral clocks in one CAN bit;
- $f_{CANCLK}$  is the Protocol Engine (PE) Clock (see Figure "CAN Engine Clocking Scheme"), in Hz;
- $f_{SYS}$  is the frequency of operation of the system (CHI) clock, in Hz;
- PSEG1 is the value in CTRL1[PSEG1] field;
- PSEG2 is the value in CTRL1[PSEG2] field;
- PROPSEG is the value in CTRL1[PROPSEG] field;
- PRES DIV is the value in CTRL1[PRES DIV] field.

For example, 180 CAN bits = 180 x NCCP peripheral clock periods.

**Table 37-106. Time segment syntax**

| Syntax         | Description                                                                                                                                    |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| SYNC_SEG       | System expects transitions to occur on the bus during this period.                                                                             |
| Transmit Point | A node in transmit mode transfers a new value to the CAN bus at this point.                                                                    |
| Sample Point   | A node samples the bus at this point. If the three samples per bit option is selected, then this point marks the position of the third sample. |

The following table gives an overview of the CAN compliant segment settings and the related parameter values.

**Table 37-107. Bosch CAN 2.0B standard compliant bit time segment settings**

| Time segment 1 | Time segment 2 | Re-synchronization jump width |
|----------------|----------------|-------------------------------|
| 5 .. 10        | 2              | 1 .. 2                        |
| 4 .. 11        | 3              | 1 .. 3                        |
| 5 .. 12        | 4              | 1 .. 4                        |
| 6 .. 13        | 5              | 1 .. 4                        |
| 7 .. 14        | 6              | 1 .. 4                        |
| 8 .. 15        | 7              | 1 .. 4                        |
| 9 .. 16        | 8              | 1 .. 4                        |

### Note

The user must ensure the bit time settings are in compliance with the CAN Protocol standard (ISO 11898-1). For bit time calculations, use an IPT (Information Processing Time) of 2, which is the value implemented in the FlexCAN module.

#### 37.5.8.5 Arbitration and matching timing

During normal reception and transmission of frames, the matching, arbitration, move-in and move-out processes are executed during certain time windows inside the CAN frame, as shown in the following figures.

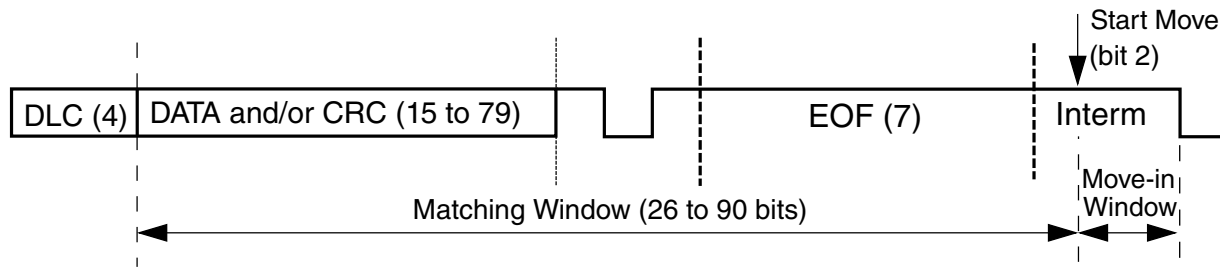


**Figure 37-94. Matching and move-in time windows**



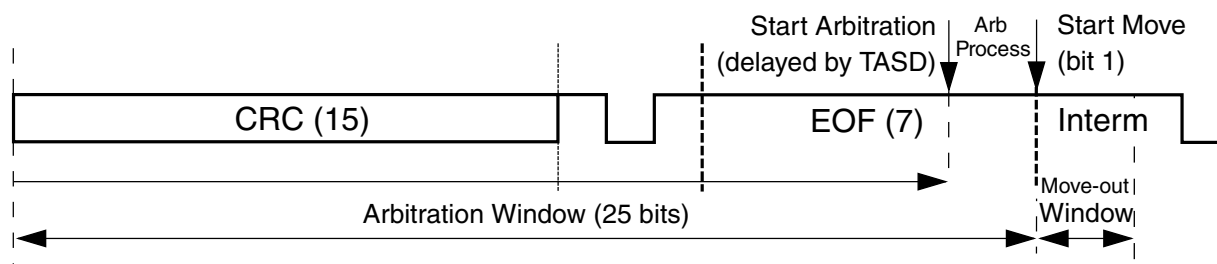


Figure 37-95. Arbitration and move-out time windows

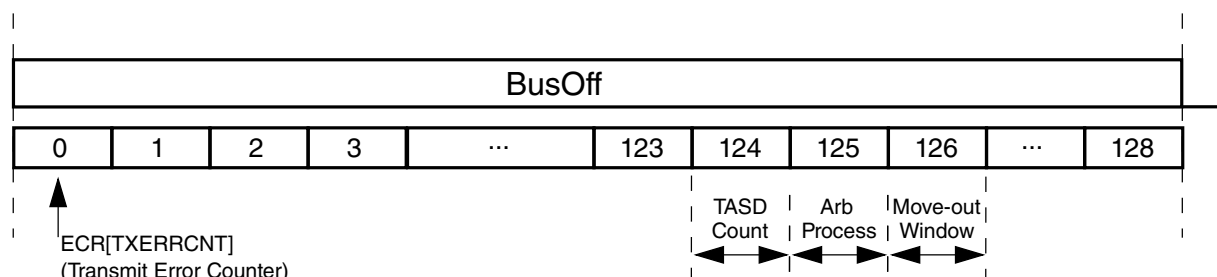


Figure 37-96. Arbitration at the end of bus off and move-out time windows

**NOTE**

The matching and arbitration timing shown in the preceding figures do not take into account the delay caused by the concurrent memory access due to the CPU or other internal FlexCAN sub-blocks.

When doing matching and arbitration, FlexCAN needs to scan the whole Message Buffer memory during the available time slot. In order to have sufficient time to do that, the following requirements must be observed:

- A valid CAN bit timing must be programmed, as indicated in [Table 37-107](#)
- The peripheral clock frequency can not be smaller than the oscillator clock frequency, see [Clock domains and restrictions](#)
- There must be a minimum ratio between the peripheral clock frequency and the CAN bit rate, as specified in the following table:

**Table 37-108. Minimum ratio between peripheral clock frequency and CAN bit rate**

| Number of Message Buffers | RFEN | Minimum number of peripheral clocks per CAN bit |
|---------------------------|------|-------------------------------------------------|
| 16 and 32                 | 0    | 16                                              |
| 64                        | 0    | 25                                              |
| 16                        | 1    | 16                                              |
| 32                        | 1    | 17                                              |

*Table continues on the next page...*

**Table 37-108. Minimum ratio between peripheral clock frequency and CAN bit rate (continued)**

| Number of Message Buffers | RFEN | Minimum number of peripheral clocks per CAN bit |
|---------------------------|------|-------------------------------------------------|
| 64                        | 1    | 30                                              |

A direct consequence of the first requirement is that the minimum number of time quanta per CAN bit must be 8, therefore the oscillator clock frequency should be at least 8 times the CAN bit rate. The minimum frequency ratio specified in the preceding table can be achieved by choosing a high enough peripheral clock frequency when compared to the oscillator clock frequency, or by adjusting one or more of the bit timing parameters (PRES DIV, PROPSEG, PSEG1, PSEG2) contained in the Control 1 Register (CTRL1).

In case of synchronous operation (when the peripheral clock frequency is equal to the oscillator clock frequency), the number of peripheral clocks per CAN bit can be adjusted by selecting an adequate value for PRES DIV in order to meet the requirement in the preceding table. In case of asynchronous operation (the peripheral clock frequency greater than the oscillator clock frequency), the number of peripheral clocks per CAN bit can be adjusted by both PRES DIV and/or the frequency ratio.

As an example, taking the case of 64 MBs, if the oscillator and peripheral clock frequencies are equal and the CAN bit timing is programmed to have 8 time quanta per bit, then the prescaler factor (PRES DIV + 1) should be at least 2. For prescaler factor equal to one and CAN bit timing with 8 time quanta per bit, the ratio between peripheral and oscillator clock frequencies should be at least 2.

### 37.5.9 Clock domains and restrictions

The FlexCAN module has two clock domains asynchronous to each other:

- The Bus Domain feeds the Control Host Interface (CHI) submodule and is derived from the peripheral clock.
- The Oscillator Domain feeds the CAN Protocol Engine (PE) submodule and is derived directly from a crystal oscillator clock, so that very low jitter performance can be achieved on the CAN bus.

When CTRL1[CLKSRC] bit is set, synchronous operation occurs because both domains are connected to the peripheral clock (creating a 1:1 ratio between the peripheral and oscillator domain clocks).

When the two domains are connected to clocks with different frequencies and/or phases, there are restrictions on the frequency relationship between the two clock domains. In the case of asynchronous operation, the Bus Domain clock frequency must always be greater than the Oscillator Domain clock frequency.

### NOTE

Asynchronous operation with a 1:1 ratio between peripheral and oscillator clocks is not allowed.

## 37.5.10 Modes of operation details

The FlexCAN module has functional modes and low-power modes. See [Modes of operation](#) for an introductory description of all the modes of operation. The following sub-sections contain functional details on Freeze mode and the low-power modes.

### CAUTION

“Permanent Dominant” failure on CAN Bus line is not supported by FlexCAN. If a Low-Power request or Freeze mode request is done during a “Permanent Dominant”, the corresponding acknowledge can never be asserted.

### 37.5.10.1 Freeze mode

This mode is requested by the CPU through the assertion of the HALT bit in the MCR Register or when the chip is put into Debug mode. In both cases it is also necessary that the FRZ bit is asserted in the MCR Register and the module is not in a low-power mode. The acknowledgement is obtained through the assertion by the FlexCAN of FRZ\_ACK bit in the same register. The CPU must only consider the FlexCAN in Freeze mode when both request and acknowledgement conditions are satisfied.

When Freeze mode is requested during transmission or reception, FlexCAN does the following:

- Waits to be in either Intermission, Passive Error, Bus Off or Idle state
- Waits for all internal activities like arbitration, matching, move-in and move-out to finish. A pending move-in is not taken into account.
- Ignores the Rx input pin and drives the Tx pin as recessive
- Stops the prescaler, thus halting all CAN protocol activities

- Grants write access to the Error Counters Register, which is read-only in other modes
- Sets the NOT\_RDY and FRZ\_ACK bits in MCR

After requesting Freeze mode, the user must wait for the FRZ\_ACK bit to be asserted in MCR before executing any other action, otherwise FlexCAN may operate in an unpredictable way. In Freeze mode, all memory mapped registers are accessible, except for CTRL1[CLK\_SRC] bit that can be read but cannot be written.

Exiting Freeze mode is done in one of the following ways:

- CPU negates the FRZ bit in the MCR Register
- The chip is removed from Debug Mode and/or the HALT bit is negated

The FRZ\_ACK bit is negated after the protocol engine recognizes the negation of the freeze request. When out of Freeze mode, FlexCAN tries to re-synchronize to the CAN bus by waiting for 11 consecutive recessive bits.

### **37.5.10.2 Module Disable mode**

This low power mode is normally used to temporarily disable a complete FlexCAN block, with no power consumption. It is requested by the CPU through the assertion of the MDIS bit in the MCR Register and the acknowledgement is obtained through the assertion by the FlexCAN of the LPM\_ACK bit in the same register. The CPU must only consider the FlexCAN in Disable mode when both request and acknowledgement conditions are satisfied.

If the module is disabled during Freeze mode, it requests to disable the clocks to the PE and CHI sub-modules, sets the LPM\_ACK bit and negates the FRZ\_ACK bit.

If the module is disabled during transmission or reception, FlexCAN does the following:

- Waits to be in either Idle or Bus Off state, or else waits for the third bit of Intermission and then checks it to be recessive
- Waits for all internal activities like arbitration, matching, move-in and move-out to finish. A pending move-in is not taken into account.
- Ignores its Rx input pin and drives its Tx pin as recessive
- Shuts down the clocks to the PE and CHI sub-modules
- Sets the NOTRDY and LPMACK bits in MCR

The Bus Interface Unit continues to operate, enabling the CPU to access memory mapped registers, except the Rx Mailboxes Global Mask Registers, the Rx Buffer 14 Mask Register, the Rx Buffer 15 Mask Register, the Rx FIFO Global Mask Register. The Rx FIFO Information Register, the Message Buffers, the Rx Individual Mask Registers, and the reserved words within RAM may not be accessed when the module is in Disable Mode. Exiting from this mode is done by negating the MDIS bit by the CPU, which causes the FlexCAN to request to resume the clocks and negate the LPM\_ACK bit after the CAN protocol engine recognizes the negation of disable mode requested by the CPU.

### 37.5.10.3 Doze mode

This is a system low power mode in which the CPU bus is kept alive and a global Doze mode request is sent to all peripherals asking them to enter low-power mode. When Doze mode is globally requested, the DOZE bit in MCR Register needs to have been asserted previously for Doze mode to be triggered. The acknowledgement is obtained through the assertion by the FlexCAN of the LPMACK bit in the same register. The CPU must only consider the FlexCAN in Doze mode when both request and acknowledgement conditions are satisfied.

If Doze mode is triggered during Freeze mode, FlexCAN requests to shut down the clocks to the PE and CHI sub-modules, sets the LPMACK bit and negates the FRZACK bit. If Doze Mode is triggered during transmission or reception, FlexCAN does the following:

- Waits to be in either Idle or Bus Off state, or else waits for the third bit of Intermission and checks it to be recessive
- Waits for all internal activities like arbitration, matching, move-in and move-out to finish. A pending move-in is not taken into account.
- Ignores its Rx input pin and drives its Tx pin as recessive
- Shuts down the clocks to the PE and CHI sub-modules
- Sets the NOTRDY and LPMACK bits in MCR

The Bus Interface Unit continues to operate, enabling the CPU to access memory mapped registers, except the Rx Mailboxes Global Mask Registers, the Rx Buffer 14 Mask Register, the Rx Buffer 15 Mask Register, the Rx FIFO Global Mask Register. The Rx FIFO Information Register, the Message Buffers, the Rx Individual Mask Registers, and the reserved words within RAM may not be accessed when the module is in Doze Mode.

Exiting Doze mode is done in one of the following ways:

- CPU removing the Doze mode request
- CPU negating the DOZE bit of the MCR Register

### **37.5.10.4 Stop mode**

This is a system low-power mode in which all chip clocks can be stopped for maximum power savings. The Stop mode is globally requested by the CPU and the acknowledgement is obtained through the assertion by the FlexCAN of a Stop Acknowledgement signal. The CPU must only consider the FlexCAN in Stop mode when both request and acknowledgement conditions are satisfied.

If FlexCAN receives the global Stop mode request during Freeze mode, it sets the LPMACK bit, negates the FRZACK bit and then sends the Stop Acknowledge signal to the CPU, in order to shut down the clocks globally.

If Stop mode is requested during transmission or reception, FlexCAN does the following:

- Waits to be in either Idle or Bus Off state, or else waits for the third bit of Intermission and checks it to be recessive
- Waits for all internal activities like arbitration, matching, move-in and move-out to finish. A pending move-in is not taken into account.
- Ignores its Rx input pin and drives its Tx pin as recessive
- Sets the NOTRDY and LPMACK bits in MCR
- Sends a Stop Acknowledge signal to the CPU, so that it can shut down the clocks globally

Stop mode is exited when the CPU resumes the clocks and removes the Stop Mode request.

After the CAN protocol engine recognizes the negation of the Stop mode request, the FlexCAN negates the LPMACK bit. FlexCAN will then wait for 11 consecutive recessive bits to synchronize to the CAN bus. As a consequence, it will not receive the frame that woke it up.

## 37.5.11 Interrupts

The module has many interrupt sources: interrupts due to message buffers and interrupts due to the ORed interrupts from MBs, Bus Off, Error, ECC Host Access Non-correctable Error, ECC FlexCAN Access Non-correctable Error, ECC Correctable Error, Tx Warning, and Rx Warning.

Each one of the message buffers can be an interrupt source, if its corresponding IMASK bit is set. There is no distinction between Tx and Rx interrupts for a particular buffer, under the assumption that the buffer is initialized for either transmission or reception. Each of the buffers has assigned a flag bit in the IFLAG registers. The bit is set when the corresponding buffer completes a successful transfer and is cleared when the CPU writes it to 1 (unless another interrupt is generated at the same time).

### Note

It must be guaranteed that the CPU clears only the bit causing the current interrupt. For this reason, bit manipulation instructions (BSET) must not be used to clear interrupt flags. These instructions may cause accidental clearing of interrupt flags which are set after entering the current interrupt service routine.

If the Rx FIFO is enabled ( $\text{MCR}[\text{RFEN}] = 1$ ), the interrupts corresponding to MBs 0 to 7 have different meanings. Bit 7 of IFLAG1 becomes the "FIFO Overflow" flag; bit 6 becomes the FIFO Warning flag, bit 5 becomes the "Frames Available in FIFO flag," and bits 4-0 are unused. See the description of IFLAG1 for more information.

For a combined interrupt where multiple MB interrupt sources are ORed together, the interrupt is generated when any of the associated MBs (or FIFO, if applicable) generates an interrupt. In this case, the CPU must read the IFLAG registers to determine which MB or FIFO source caused the interrupt.

The interrupt sources for Bus Off, Error, Tx Warning, and Rx Warning generate interrupts like the MB interrupt sources, and they can be read from ESR1 and ESR2. The Bus Off, Error, Tx Warning, and Rx Warning interrupt mask bits are located in CTRL1.

The interrupt sources for ECC errors—Host Access with Non-correctable Interrupt Flag, FlexCAN Access with Non-correctable Interrupt Flag, and Correctable Error Interrupt Flag—can be read in ERRSR and the corresponding interrupts masks bits are located in MECR.

## 37.5.12 Bus interface

The CPU access to FlexCAN registers are subject to the following rules:

- Unrestricted read and write access to supervisor registers (registers identified with S/U in Table "Module Memory Map" in Supervisor Mode or with S only) results in access error.
- Read and write access to implemented reserved address space results in access error.
- Write access to positions whose bits are all currently read-only results in access error. If at least one of the bits is not read-only then no access error is issued. Write permission to positions or some of their bits can change depending on the mode of operation or transitory state. Refer to register and bit descriptions for details.
- Read and write access to unimplemented address space results in access error.
- Read and write access to RAM located positions during Low Power Mode results in access error.
- If MAXMB is programmed with a value smaller than the available number of MBs, then the unused memory space can be used as general purpose RAM space. Note that reserved words within RAM cannot be used. As an example, suppose FlexCAN is configured with 16 MBs, RFFN is 0x0, and MAXMB is programmed with zero. The maximum number of MBs in this case becomes one. The RAM starts at 0x0080, and the space from 0x0080 to 0x008F is used by the one MB. The memory space from 0x0090 to 0x017F is available. The space between 0x0180 and 0x087F is reserved. The space from 0x0880 to 0x0883 is used by the one Individual Mask and the available memory in the Mask Registers space would be from 0x0884 to 0x08BF. From 0x08C0 through 0x09DF there are reserved words for internal use which cannot be used as general purpose RAM. As a general rule, free memory space for general purpose depends only on MAXMB.

## 37.5.13 Detection and Correction of Memory Errors

FlexCAN supports detection and correction of errors in memory read accesses. Each byte of FlexCAN memory is associated to 5 parity bits and the error correction mechanism assures that in this 13-bit word, errors in one bit can be corrected (corrected errors) and errors in 2 bits can be detected but not corrected (non-corrected errors). Errors in more than 2 bits may not be detected. When read, the parity bits are used to calculate a syndrome, which indicates the error in each byte.



Memory errors are indicated to Host through status register (Error Status Register (ERRSR)) and bus transfer errors, and reported through report registers (Error Report Address Register (RERRAR), Error Report Data Register (RERRDR) and Error Report Syndrome Register (RERRSYNR)).

The error detection and correction mechanism can be activated or not, controlled by ECCDIS bit in Memory Error Control Register (MECR). When disabled, updates on indications and reporting registers are stopped, but the parity bits are still calculated and written along with data in memory write operations. It assures that memory has consistent parity bits associated to data.

### NOTE

All FlexCAN memory must be initialized before starting its operation in order to have the parity bits in memory properly updated. The WRMFRZ bit in Control 2 Register (CTRL2) grants write access to all memory positions from 0x080 to 0xADF.

To avoid that critical error correction configuration be accidentally changed, this protocol must be followed to enable the update of the Memory Error Control Register (MECR):

1. By default, ECRWRE bit in Control 2 Register (CTRL2) is 0 and ECRWRDIS bit in Memory Error Control Register (MECR) is 1.
2. Set ECRWRE bit in Control 2 Register (CTRL2).
3. Clear ECRWRDIS bit.
4. All writes to Memory Error Control Register (MECR) must keep ECRWRDIS cleared.
5. After configuration is done, lock the Memory Error Control Register (MECR) by either setting ECRWRDIS or clearing ECRWRE.

### 37.5.13.1 Sources of the Memory Access

The FlexCAN memory can be accessed by 2 major sources (or requestors):

- by Host (CPU): largest word accessed is 32-bit wide
- by FlexCAN internals processes (Match, Arbitration, RxMove on reception, TxMove on transmission): largest word accessed is 64-bit wide

The way that non-correctable errors are indicated and reported depends on the source of access.

### 37.5.13.2 Error Indication

Memory errors are indicated by flags HANCEIF, FANCEIF, CEIF in the Error Status Register (ERRSR). Non-correctable errors detected in memory reads requested by Host are indicated separately than the ones detected in requests by FlexCAN internal processes. Corrected errors indication makes no distinction of the source of the access. There are 3 independent flags for these 3 cases, and each flag will raise an interrupt unless it is masked by mask bits in Memory Error Control Register (MECR). If both non-corrected and corrected errors are found in different bytes in the same read operation both flags are set.

The non-corrected error detected in Host accesses are also indicated as a bus transfer error. A bus wait request may be asserted to extend the memory transaction to the moment the report registers are updated. This indication cannot be masked. If the flag bit HANCEIF is not masked, the same non-corrected error will raise a bus transfer error and an interrupt request.

Each indication flag has one Overrun flag in Error Status Register (ERRSR). The Overrun flags do not request interrupts. Overrun flags for non-correctable errors indicate that other errors of the same nature were detected after current error being treated, while overrun flags for correctable errors indicate that other errors of the same nature were detected before the current error being treated. This is the recommended handling sequence for error indication:

1. Get error report information from report registers
2. Use this information to take proper measures in the application
3. Clear the Interrupt flag
4. If the Overrun flag is active:
  - a. Alert application that at least one error could not be handled
  - b. Clear the Overrun flag

The FlexCAN internal processes can access memory in transactions larger than 32-bits. For the indication, this kind of access is considered a consecutive sequence of 32-bit accesses. If errors are found in 2 or more 32-bit words the Interrupt and Overrun flags are set simultaneously.

### 37.5.13.3 Error Reporting

Report registers Error Report Address Register (RERRAR), Error Report Data Register (RERRDR) and Error Report Syndrome Register (RERRSYNR) provide detailed information about the address read, raw data and syndrome read with error and indicated by the flags described in the “Error Indication” section. The address, data and syndrome registers are updated simultaneously along with the error flags, according to these rules.

1. If any of the 2 non-corrected error flags is currently set, does not update the registers (preserve the old non-corrected error reporting)
2. Else (no error flag is currently set or only the corrected error flag is currently set), update the report registers according to the new error; or according to the most severe of new errors if non-corrected and corrected errors are simultaneously detected

Reporting of errors detected in accesses larger than 32-bit follows the semantic described in the “Error Indication” section and in the Error Report Address Register (RERRAR) description.

The address reported in RERRAR and defined in ERRIAR are not the same listed in the module memory map. How the reported addresses correspond to locations in the module memory map is shown in the Error Injection Address Register (ERRIAR) description.

Addresses reported when reading memory portions organized as FIFOs, such as the Rx FIFO Structure and the Rx FIFO Information Register (RXFIR), refer to the address of the element accessed in the FIFO, not to the FIFO base address.

To assure coherence of the error report registers it is necessary to turn off the report update by clearing the MECR[RERRDIS] bit before reading the report registers.

### 37.5.13.4 Response to Errors

Correctable errors have no consequence on FlexCAN operation, once the corrected data is used by Host or FlexCAN internal processes. For host-initiated reads, a non-corrected error detected does not affect FlexCAN operation.

Non-corrected errors detected on memory reads requested by FlexCAN do not stop the FlexCAN processing and this may result in incorrect operation.

If a non-corrected error occurs during a reception process (Match or RxMove), an incorrect destination may be selected to store the incoming frame, a corrupted frame may be stored in the correct destination, or both. If a non-corrected error occurs during either an Arbitration or a TxMove process, either a non-highest priority frame may be mistakenly selected to be transmitted or no frame may be transmitted at all, when MECR[NCEFAFRZ] = 1.

The error report registers can provide information to the application for a customized handling of these situations. If the delay of this handling is not accepted and MECR[NCEFAFRZ] = 1, the FlexCAN stops operation automatically and enters Freeze mode when these errors are detected. See the Memory Error Control Register (MECR) description for more details.

### 37.5.13.5 Error Injection

The error injection registers ERRIAR, ERRIDPR, and ERRIPPR are used to inject errors in memory reads in order to force errors and consequently update the indication and reporting registers. How the error injection addresses correspond to locations in the module memory map is shown in the ERRIAR description.

The injection is done by flipping the data and parity bits correspondent to the bits in 1 in ERRIDPR and ERRIPPR. Injection can be selected specifically for memory accesses requested by Host or by FlexCAN internal processes.

In case of accesses larger than 32-bits, the EXTERRIE bit in Memory Error Control Register (MECR) extends the injection pattern, replicating it in 32-bit words to fill the width of the access.

#### NOTE

It is very unlikely, but error injection may correct a bit with error. This will not raise the error flags and reports as expected.

To ensure coherence among error injection registers and avoid spurious error injections, the HAERRIE and FAERRIE bits in Memory Error Control Register (MECR) must be cleared while configuring the memory injection registers.

## 37.6 Initialization/application information

This section provide instructions for initializing the FlexCAN module.

### 37.6.1 FlexCAN initialization sequence

The FlexCAN module may be reset as follows:

- Chip level hard reset, which resets all memory mapped registers asynchronously
- SOFTRST bit in MCR, which resets some of the memory mapped registers synchronously. See [Table 37-2](#) to see what registers are affected by soft reset.

Soft reset is synchronous and has to follow an internal request/acknowledge procedure across clock domains. Therefore, it may take some time to fully propagate its effects. The SOFTRST bit remains asserted while soft reset is pending, so software can poll this bit to

know when the reset has completed. Also, soft reset can not be applied while clocks are shut down in a low power mode. The low power mode should be exited and the clocks resumed before applying soft reset.

The clock source (CLKSRC bit) should be selected while the module is in Disable mode. After the clock source is selected and the module is enabled (MDIS bit negated), FlexCAN automatically goes to Freeze mode. In Freeze mode, FlexCAN is unsynchronized to the CAN bus, the HALT and FRZ bits in MCR Register are set, the internal state machines are disabled and the FRZACK and NOTRDY bits in the MCR Register are set. The Tx pin is in recessive state and FlexCAN does not initiate any transmission or reception of CAN frames. Note that the Message Buffers and the Rx Individual Mask Registers are not affected by reset, so they are not automatically initialized.

For any configuration change/initialization it is required that FlexCAN is put into Freeze mode; see [Freeze mode](#). The following is a generic initialization sequence applicable to the FlexCAN module:

- Initialize the Module Configuration Register
  - Enable the individual filtering per MB and reception queue features by setting the IRMQ bit
  - Enable the warning interrupts by setting the WRNEN bit
  - If required, disable frame self reception by setting the SRXDIS bit
  - Enable the Rx FIFO by setting the RFEN bit
  - Enable the abort mechanism by setting the AEN bit
  - Enable the local priority feature by setting the LPRIOEN bit
- Initialize the Control Register
  - Determine the bit timing parameters: PROPSEG, PSEG1, PSEG2, RJW
  - Determine the bit rate by programming the PRESDIV field
  - Determine the internal arbitration mode (LBUF bit)
- Initialize the Message Buffers
  - The Control and Status word of all Message Buffers must be initialized
  - If Rx FIFO was enabled, the ID filter table must be initialized
  - Other entries in each Message Buffer should be initialized as required

- Initialize the Rx Individual Mask Registers
- When ECC is enabled, follow the initialization note in [Detection and Correction of Memory Errors](#)
- Set required interrupt mask bits in the IMASK Registers (for all MB interrupts) and in CTRL Register (for Bus Off and Error interrupts)
- Negate the HALT bit in MCR

Starting with the last event, FlexCAN attempts to synchronize to the CAN bus.

## **Chapter 38**

# **Modular CAN (M\_CAN)**

### **38.1 Chip-specific M\_CAN information**

#### **38.1.1 M\_CAN Message RAM allocation**

On this chip, each M\_CAN instance can address 1216 words in the Message RAM.

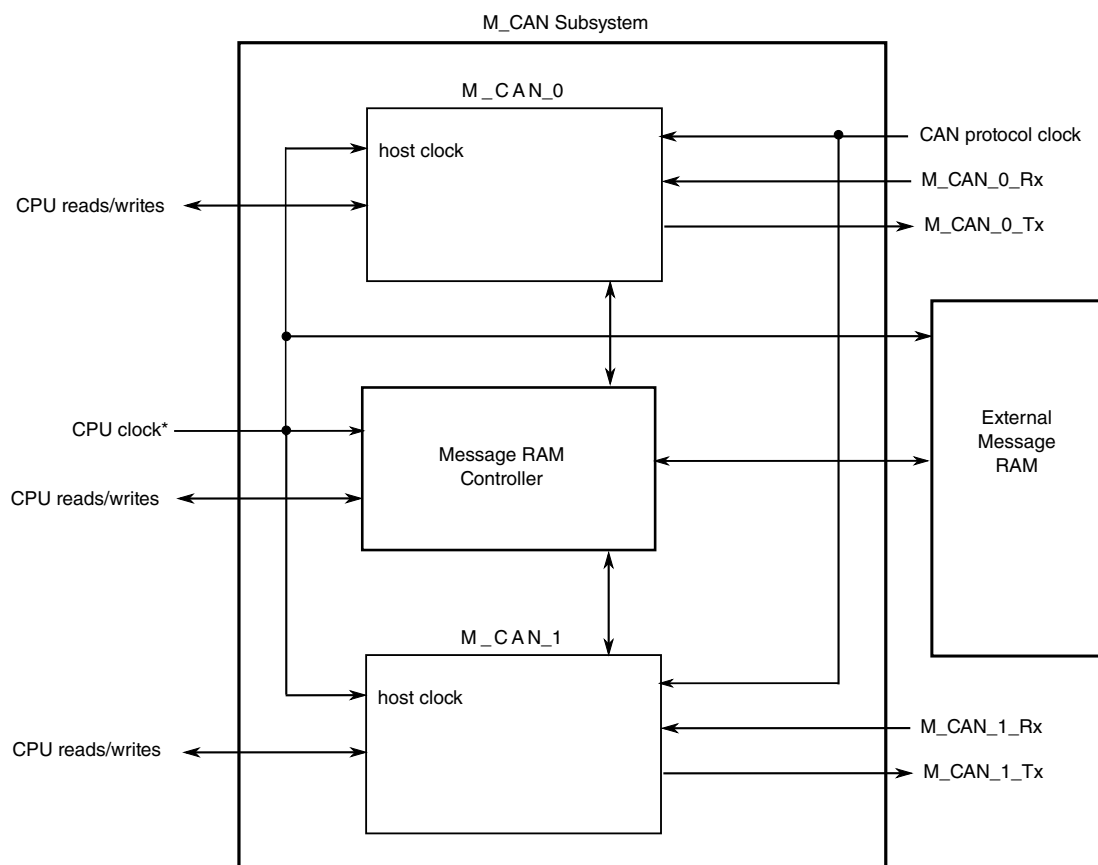
As a result, the Message RAM shared by the two M\_CAN instances supports 2432 words, or 9.5 KB.

#### **38.1.2 Introduction**

The M\_CAN subsystem includes:

- Two M\_CAN modules
- A Message RAM controller

The M\_CAN subsystem block diagram is shown in the following figure.



\*Refer to the Clocking chapter for M\_CAN clock details.

**Figure 38-1. M\_CAN subsystem block diagram**

## 38.1.3 Functional Description

### 38.1.3.1 Message RAM Controller

The Message RAM Controller has the arbiter for the accesses to the external Message RAM and the ECC (Error Code Correction) Controller for the external Message RAM data.

#### 38.1.3.1.1 Message RAM Arbiter

The Message RAM Arbiter is a dynamic round robin arbiter that selects which request is sent to the external Message RAM. These requests are made by the CPU, M\_CAN\_0, or M\_CAN\_1.

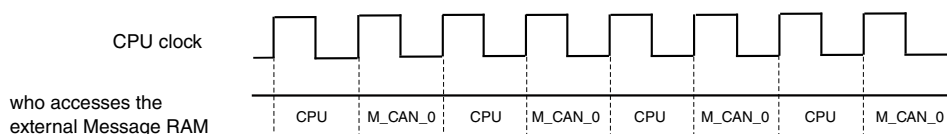
This arbiter ensures:



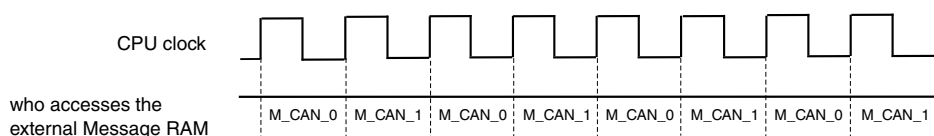
- 50% bandwidth for CPU accesses to the external Message RAM. The CPU does not wait for more than one clock cycle to access the external Message RAM (see examples 1 and 3).
- 50% bandwidth is shared between M\_CAN\_0 and M\_CAN\_1 accesses to the external Message RAM. Each M\_CAN waits at least one clock cycle to access the external Message RAM (see examples 1, 2, and 3).
- If there is no CPU request, all bandwidth is distributed to M\_CAN\_0 and M\_CAN\_1 (see example 2).
- If there are no M\_CAN\_0 and M\_CAN\_1 requests, all bandwidth is distributed to the CPU.
- If there are requests from only one M\_CAN, the other M\_CAN's bandwidth is distributed to the first M\_CAN (see example 1).

The following examples illustrate the dynamic arbiter scheme.

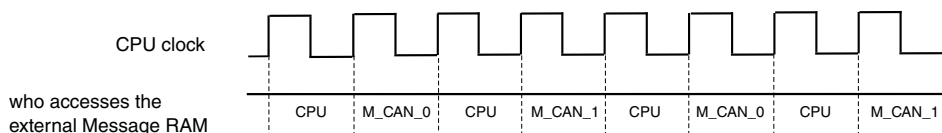
- **Example 1:** The following figure shows who accesses the external Message RAM when only CPU and M\_CAN\_0 try to access to it.



- **Example 2:** The following figure shows who accesses the external Message RAM when only M\_CAN\_0 and M\_CAN\_1 try to access to it.



- **Example 3:** The following figure shows who accesses the external Message RAM when the CPU, M\_CAN\_0, and M\_CAN\_1 try to access to it.



The read or write accesses to the external Message RAM use two clock cycles. In the first clock cycle, the address is available, and in the second, the data is available.

The arbiter has a pseudo address pre-fetching mechanism that allows the data of the previous access to overlap with the address of the current access. The pseudo address pre-fetching scheme saves multiple clock cycles when there are multiples accesses to the external Message RAM.

#### **38.1.3.1.2 ECC Controller**

The ECC Controller provides Single Error Correction / Double Error Detection (SECEDED). It guarantees single bit error correction and double bit error detection (without correction). The SECEDED code is not guaranteed to detect more than two bits with error.

Each 32 data bits of the external Message RAM is associated with 7 ECC bits. If all these 39 bits are zero or one, then it is flagged as non-correctable error.

For writes to the external Message RAM, the ECC bits (7-bit) are calculated using the data bits (32-bit). The data bits plus ECC bits (39-bit) are written into the specified memory address. The error detection and correction are performed on the reads from the external Message RAM.

When an M\_CAN accesses the external Message RAM, the ECC bits are calculated by the ECC Controller and they are sent to this M\_CAN.

#### **38.1.3.2 External Message RAM**

The external Message RAM supports only 32-bit write and read accesses.

The CPU can access the external Message RAM through the M\_CAN subsystem. In this case, the CPU can do 8/16/32-bit read accesses to the external Message RAM.

#### **38.1.3.3 Transfer Error**

The M\_CAN subsystem does not report any transfer error.

### **38.1.4 External Signals**

The M\_CAN subsystem external signals are shown in the following table.

**Table 38-1. M\_CAN subsystem external signals**

| Signal name | Direction | Description           |
|-------------|-----------|-----------------------|
| M_CAN_0_Rx  | input     | M_CAN_0 CAN Rx signal |
| M_CAN_0_Tx  | output    | M_CAN_0 CAN Tx signal |
| M_CAN_1_Rx  | input     | M_CAN_1 CAN Rx signal |
| M_CAN_1_Tx  | output    | M_CAN_1 CAN Tx signal |

## 38.2 Overview

The M\_CAN module is the new CAN Communication Controller IP-module that can be integrated as stand-alone device or as part of an ASIC. It is described in VHDL on RTL level, prepared for synthesis. The M\_CAN performs communication according to ISO11898-1 (Bosch CAN specification 2.0 part A,B) and to Bosch CAN FD specification V1.0. Additional transceiver hardware is required for connection to the physical layer.

The message storage is intended to be a single- or dual-ported Message RAM outside of the module. It is connected to the M\_CAN via the Generic Master Interface. Depending on the chosen ASIC integration, multiple M\_CAN controllers can share the same Message RAM.

All functions concerning the handling of messages are implemented by the Rx Handler and the Tx Handler. The Rx Handler manages message acceptance filtering, the transfer of received messages from the CAN Core to the Message RAM as well as providing receive message status information. The Tx Handler is responsible for the transfer of transmit messages from the Message RAM to the CAN Core as well as providing transmit status information.

Acceptance filtering is implemented by a combination of up to 128 filter elements where each one can be configured as a range, as a bit mask, or as a dedicated ID filter.

The M\_CAN can be connected to a wide range of Host CPUs via its 8/16/32-bit Generic Slave Interface. The M\_CAN's clock domain concept allows the separation between the high precision CAN clock and the Host clock, which may be generated by an FM-PLL.

### 38.2.1 Features

The following are the features of M\_CAN.

- Conforms with CAN protocol version 2.0 part A, B and ISO 11898-1

- CAN FD with up to 64 data bytes supported
- CAN Error Logging
- AUTOSAR support
- SAE J1939 support
- Improved acceptance filtering
- Two configurable Receive FIFOs
- Separate signalling on reception of High Priority Messages
- Up to 64 dedicated Receive Buffers
- Up to 32 dedicated Transmit Buffers
- Configurable Transmit FIFO
- Configurable Transmit Queue
- Configurable Transmit Event FIFO
- Direct Message RAM access for Host CPU
- Multiple M\_CANs may share the same Message RAM
- Programmable loop-back test mode
- Maskable module interrupts
- 8/16/32 bit Generic Slave Interface for connection customer-specific Host CPUs
- Two clock domains (CAN clock and Host clock)
- Power-down support
- Debug on CAN support

## 38.2.2 Block Diagram

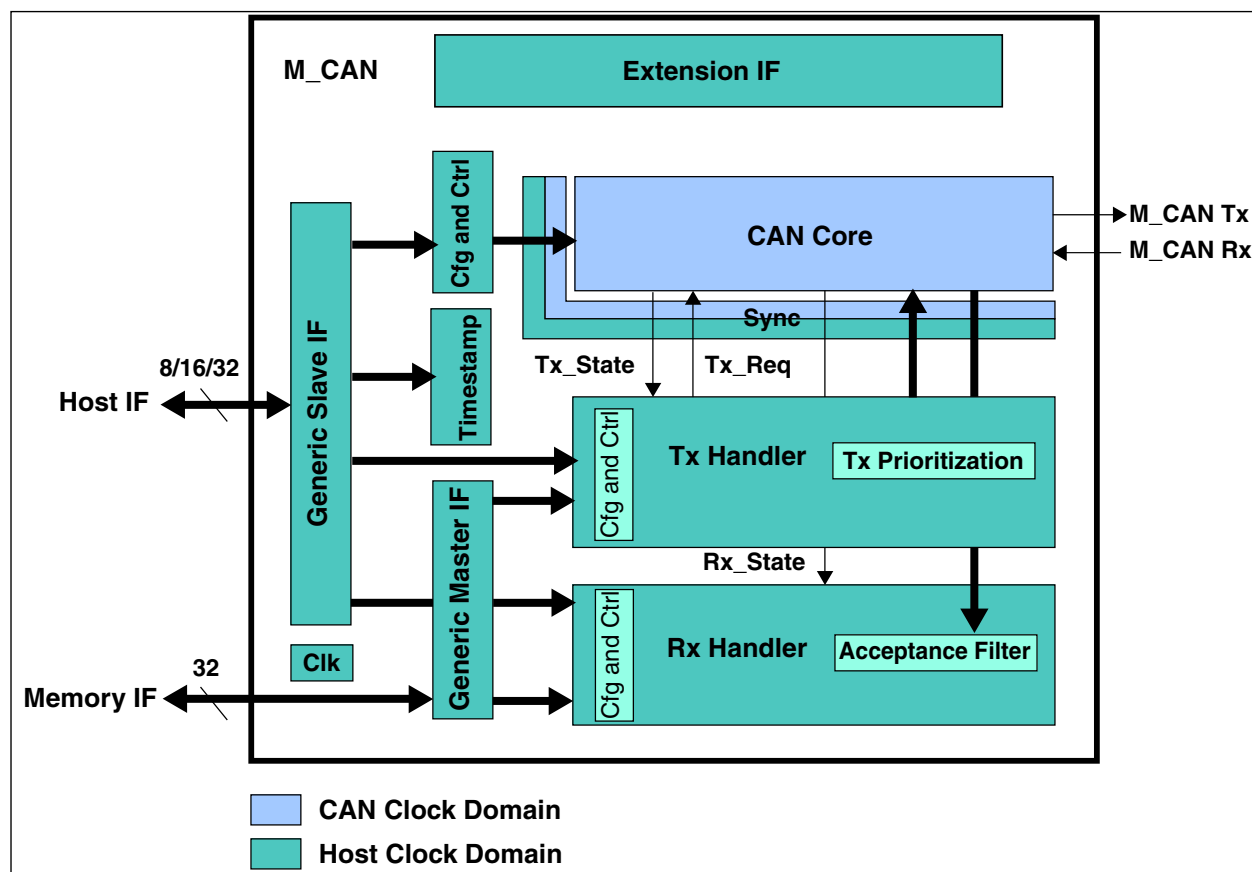


Figure 38-2. M\_CAN Block Diagram

- **CAN Core:** CAN Protocol Controller and Rx/Tx Shift Register. Handles all ISO 11898-1 protocol functions. Supports 11-bit and 29-bit identifiers.
- **Sync:** Synchronizes signals from the Host clock domain to the CAN clock domain and vice versa.
- **Clk:** Synchronizes reset signal to the Host clock domain and to the CAN clock domain.
- **Cfg and Ctrl:** CAN Core related configuration and control bits.
- **Interrupt and Timestamp:** Interrupt control and 16-bit CAN bit time counter for receive and transmit timestamp generation.
- **Tx Handler:** Controls the message transfer from the external Message RAM to the CAN Core. A maximum of 32 Tx Buffers can be configured for transmission. Tx buffers can be used as dedicated Tx Buffers, as Tx FIFO, part of a Tx Queue, or as a combination of them. A Tx Event FIFO stores Tx timestamps together with the corresponding Message ID. Transmit cancellation is also supported.

- **Rx Handler:** Controls the transfer of received messages from the CAN Core to the external Message RAM. The Rx Handler supports two Receive FIFOs, each of configurable size, and up to 64 dedicated Rx Buffers for storage of all messages that have passed acceptance filtering. A dedicated Rx Buffer, in contrast to a Receive FIFO, is used to store only messages with a specific identifier. An Rx timestamp is stored together with each message. Up to 128 filters can be defined for 11-bit IDs and up to 64 filters for 29-bit IDs.
- Connects the M\_CAN to a customer specific Host CPU. The Generic Slave Interface is capable to connect to an 8/16/32-bit bus to support a wide range of interconnection structures.
- **Generic Master Interface:** Connects the M\_CAN access to an external 32-bit Message RAM. The maximum Message RAM size is 16 KB × 32-bit. A single M\_CAN can use at most 4.25K × 32 bit.
- All flags from the Interrupt Register IR as well as selected internal status and control signals are routed to this interface. The interface is intended for connection of the M\_CAN to a module-external interrupt unit or to other module-external components. The connection of these signals is optional.

### 38.2.3 Dual Clock Sources

To improve the EMC behavior, a spread spectrum clock can be used for the Host clock domain. Due to the high precision clocking requirements of the CAN Core, a separate clock without any modulation has to be provided as CAN clock.

Within the M\_CAN module there is a synchronization mechanism implemented to ensure save data transfer between the two clock domains.

#### Note

In order to achieve a stable function of the M\_CAN, the Host clock must always be faster than or equal to the CAN clock. Also, the modulation depth of a spread spectrum clock has to be regarded.

## 38.2.4 Dual Interrupt Lines

The module provides two interrupt lines. Interrupts can be routed either to M\_CAN interrupt 0 or to M\_CAN interrupt 1. By default all interrupts are routed to interrupt line M\_CAN interrupt 0. By programming ILE[EINT0] and ILE[EINT1], the interrupt lines can be enabled or disabled separately.

## 38.3 Memory Map and Register Description

After hardware reset, the registers of the M\_CAN hold the reset values listed in Table 1. Additionally the Bus\_Off state is reset and the output M\_CAN Tx is set to recessive (HIGH). The value 0x0001 (CCCR[INIT] = 1) in the CC Control Register enables software initialization. The M\_CAN does not influence the CAN bus until the CPU resets CCCR[INIT] to 0.

The M\_CAN module allocates an address space of 256 bytes. All registers are organized as 32-bit registers. The M\_CAN is accessible by the Host CPU via the Generic Slave Interface using a data width of 8 bit (byte access), 16 bit (half-word access), or 32 bit (word access). Write access by the Host CPU to registers/bits marked with “P=Protected Write” is possible only with CCCR[CCE]=1 AND CCCR[INIT]=1. There is a delay from writing to a command register until the update of the related status register bits due to clock domain crossing.

### CAUTION

In case the application software wants to access one of the reserved addresses in the M\_CAN register map (read or write access), interrupt flag IR[ARA] is set, and if enable the interrupt is signalled via the assigned interrupt line (m\_can\_int0 or m\_can\_int1).

**M\_CAN memory map**

| Address offset (hex) | Register name                                          | Width (in bits) | Access | Reset value                 | Section/ page               |
|----------------------|--------------------------------------------------------|-----------------|--------|-----------------------------|-----------------------------|
| 0                    | Core Release Register (M_CAN_CREL)                     | 32              | R      | <a href="#">See section</a> | <a href="#">38.3.1/1773</a> |
| 4                    | Endian Register (M_CAN_ENDN)                           | 32              | R      | 8765_4321h                  | <a href="#">38.3.2/1774</a> |
| C                    | Data Bit Timing and Prescaler Register (M_CAN_DBTP)    | 32              | R/W    | 0000_0A33h                  | <a href="#">38.3.3/1775</a> |
| 10                   | Test Register (M_CAN_TEST)                             | 32              | R/W    | <a href="#">See section</a> | <a href="#">38.3.4/1776</a> |
| 14                   | RAM Watchdog Register (M_CAN_RWD)                      | 32              | R/W    | 0000_0000h                  | <a href="#">38.3.5/1778</a> |
| 18                   | CC Control Register (M_CAN_CCCR)                       | 32              | R/W    | 0000_0001h                  | <a href="#">38.3.6/1779</a> |
| 1C                   | Nominal Bit Timing and Prescaler Register (M_CAN_NBTP) | 32              | R/W    | 0600_0A03h                  | <a href="#">38.3.7/1781</a> |

*Table continues on the next page...*

**M\_CAN memory map (continued)**

| Address offset (hex) | Register name                                           | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|---------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 20                   | Timestamp Counter Configuration Register (M_CAN_TSCC)   | 32              | R/W    | 0000_0000h  | <a href="#">38.3.8/1783</a>  |
| 24                   | Timestamp Counter Value Register (M_CAN_TSCV)           | 32              | w1c    | 0000_0000h  | <a href="#">38.3.9/1784</a>  |
| 28                   | Timeout Counter Configuration Register (M_CAN_TOCC)     | 32              | R/W    | FFFF_0000h  | <a href="#">38.3.10/1784</a> |
| 2C                   | Timeout Counter Value Register (M_CAN_TOCV)             | 32              | w1c    | 0000_FFFFh  | <a href="#">38.3.11/1785</a> |
| 40                   | Error Counter Register (M_CAN_ECR)                      | 32              | R      | 0000_0000h  | <a href="#">38.3.12/1786</a> |
| 44                   | Protocol Status Register (M_CAN_PSR)                    | 32              | R      | 0000_0707h  | <a href="#">38.3.13/1787</a> |
| 48                   | Transmitter Delay Compensation Register (M_CAN_TDCR)    | 32              | R/W    | 0000_0000h  | <a href="#">38.3.14/1790</a> |
| 50                   | Interrupt Register (M_CAN_IR)                           | 32              | R/W    | 0000_0000h  | <a href="#">38.3.15/1791</a> |
| 54                   | Interrupt Enable Register (M_CAN_IE)                    | 32              | R/W    | 0000_0000h  | <a href="#">38.3.16/1794</a> |
| 58                   | Interrupt Line Select Register (M_CAN_ILS)              | 32              | R/W    | 0000_0000h  | <a href="#">38.3.17/1797</a> |
| 5C                   | Interrupt Line Enable Register (M_CAN_ILE)              | 32              | R/W    | 0000_0000h  | <a href="#">38.3.18/1801</a> |
| 80                   | Global Filter Configuration Register (M_CAN_GFC)        | 32              | R/W    | 0000_0000h  | <a href="#">38.3.19/1802</a> |
| 84                   | Standard ID Filter Configuration Register (M_CAN_SIDFC) | 32              | R/W    | 0000_0000h  | <a href="#">38.3.20/1803</a> |
| 88                   | Extended ID Filter Configuration Register (M_CAN_XIDFC) | 32              | R/W    | 0000_0000h  | <a href="#">38.3.21/1804</a> |
| 90                   | Extended ID and Mask Register (M_CAN_XIDAM)             | 32              | R/W    | 1FFF_FFFFh  | <a href="#">38.3.22/1804</a> |
| 94                   | High Priority Message Status Register (M_CAN_HPMS)      | 32              | R      | 0000_0000h  | <a href="#">38.3.23/1805</a> |
| 98                   | New Data 1 Register (M_CAN_NDAT1)                       | 32              | R/W    | 0000_0000h  | <a href="#">38.3.24/1806</a> |
| 9C                   | New Data 2 Register (M_CAN_NDAT2)                       | 32              | R/W    | 0000_0000h  | <a href="#">38.3.25/1806</a> |
| A0                   | Rx FIFO 0 Configuration Register (M_CAN_RXF0C)          | 32              | R/W    | 0000_0000h  | <a href="#">38.3.26/1807</a> |
| A4                   | Rx FIFO 0 Status Register (M_CAN_RXF0S)                 | 32              | R      | 0000_0000h  | <a href="#">38.3.27/1808</a> |
| A8                   | Rx FIFO 0 Acknowledge Register (M_CAN_RXF0A)            | 32              | R/W    | 0000_0000h  | <a href="#">38.3.28/1809</a> |
| AC                   | Rx Buffer Configuration Register (M_CAN_RXBC)           | 32              | R/W    | 0000_0000h  | <a href="#">38.3.29/1809</a> |
| B0                   | Rx FIFO 1 Configuration Register (M_CAN_RXF1C)          | 32              | R/W    | 0000_0000h  | <a href="#">38.3.30/1810</a> |

Table continues on the next page...



**M\_CAN memory map (continued)**

| Address offset (hex) | Register name                                                            | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|--------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| B4                   | Rx FIFO 1 Status Register (M_CAN_RXF1S)                                  | 32              | R      | 0000_0000h  | <a href="#">38.3.31/1811</a> |
| B8                   | Rx FIFO 1 Acknowledge Register (M_CAN_RXF1A)                             | 32              | R/W    | 0000_0000h  | <a href="#">38.3.32/1812</a> |
| BC                   | Rx Buffer / FIFO Element Size Configuration Register (M_CAN_RXESC)       | 32              | R/W    | 0000_0000h  | <a href="#">38.3.33/1813</a> |
| C0                   | Tx Buffer Configuration Register (M_CAN_TXBC)                            | 32              | R/W    | 0000_0000h  | <a href="#">38.3.34/1815</a> |
| C4                   | Tx FIFO/Queue Status Register (M_CAN_TXFQS)                              | 32              | R      | 0000_0000h  | <a href="#">38.3.35/1816</a> |
| C8                   | Tx Buffer Element Size Configuration (M_CAN_TXESC)                       | 32              | R/W    | 0000_0000h  | <a href="#">38.3.36/1817</a> |
| CC                   | Tx Buffer Request Pending Register (M_CAN_TXBRP)                         | 32              | R      | 0000_0000h  | <a href="#">38.3.37/1818</a> |
| D0                   | Tx Buffer Add Request Register (M_CAN_TXBAR)                             | 32              | R/W    | 0000_0000h  | <a href="#">38.3.38/1819</a> |
| D4                   | Tx Buffer Cancellation Request Register (M_CAN_TXBCR)                    | 32              | R/W    | 0000_0000h  | <a href="#">38.3.39/1819</a> |
| D8                   | Tx Buffer Transmission Occurred Register (M_CAN_TXBTO)                   | 32              | R      | 0000_0000h  | <a href="#">38.3.40/1820</a> |
| DC                   | Tx Buffer Cancellation Finished Register (M_CAN_TXBCF)                   | 32              | R      | 0000_0000h  | <a href="#">38.3.41/1820</a> |
| E0                   | Tx Buffer Transmission Interrupt Enable Register (M_CAN_TXBTIE)          | 32              | R/W    | 0000_0000h  | <a href="#">38.3.42/1821</a> |
| E4                   | Tx Buffer Cancellation Finished Interrupt Enable Register (M_CAN_TXBCIE) | 32              | R/W    | 0000_0000h  | <a href="#">38.3.43/1821</a> |
| F0                   | Tx Event FIFO Configuration Register (M_CAN_TXEFC)                       | 32              | R/W    | 0000_0000h  | <a href="#">38.3.44/1822</a> |
| F4                   | Tx Event FIFO Status Register (M_CAN_TXEFS)                              | 32              | R      | 0000_0000h  | <a href="#">38.3.45/1823</a> |
| F8                   | Tx Event FIFO Acknowledge Register (M_CAN_TXEFA)                         | 32              | R/W    | 0000_0000h  | <a href="#">38.3.46/1824</a> |

**38.3.1 Core Release Register (M\_CAN\_CREL)**

The following table shows example field values for this register and explains how they encode a particular M\_CAN core release.

| REL | STEP | SUBSTEP | YEAR | MON | DAY | Release                         |
|-----|------|---------|------|-----|-----|---------------------------------|
| 0   | 2    | 0       | 9    | 3   | 26  | Revision 0.2.0, date 2009/03/26 |

## Memory Map and Register Description

Address: 0h base + 0h offset = 0h

|       |     |    |    |    |      |    |    |    |         |    |    |    |      |    |    |    |     |    |    |    |    |    |    |    |     |    |    |    |    |    |    |    |
|-------|-----|----|----|----|------|----|----|----|---------|----|----|----|------|----|----|----|-----|----|----|----|----|----|----|----|-----|----|----|----|----|----|----|----|
| Bit   | 0   | 1  | 2  | 3  | 4    | 5  | 6  | 7  | 8       | 9  | 10 | 11 | 12   | 13 | 14 | 15 | 16  | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24  | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | REL |    |    |    | STEP |    |    |    | SUBSTEP |    |    |    | YEAR |    |    |    | MON |    |    |    |    |    |    |    | DAY |    |    |    |    |    |    |    |
| W     |     |    |    |    |      |    |    |    |         |    |    |    |      |    |    |    |     |    |    |    |    |    |    |    |     |    |    |    |    |    |    |    |
| Reset | x*  | x* | x* | x* | x*   | x* | x* | x* | x*      | x* | x* | x* | x*   | x* | x* | x* | x*  | x* | x* | x* | x* | x* | x* | x* | x*  | x* | x* | x* | x* | x* | x* | x* |

\* Notes:

- x = Undefined at reset.

### M\_CAN\_CREL field descriptions

| Field           | Description                                                                                           |
|-----------------|-------------------------------------------------------------------------------------------------------|
| 0–3<br>REL      | Core Release<br>One digit, BCD-coded.                                                                 |
| 4–7<br>STEP     | Step of Core Release<br>One digit, BCD-coded.                                                         |
| 8–11<br>SUBSTEP | Sub-step of Core Release<br>One digit, BCD-coded.                                                     |
| 12–15<br>YEAR   | Time Stamp Year<br>One digit, BCD-coded. This field is set by generic parameter on M_CAN synthesis.   |
| 16–23<br>MON    | Time Stamp Month<br>Two digits, BCD-coded. This field is set by generic parameter on M_CAN synthesis. |
| 24–31<br>DAY    | Time Stamp Day<br>Two digits, BCD-coded. This field is set by generic parameter on M_CAN synthesis.   |

## 38.3.2 Endian Register (M\_CAN\_ENDN)

Address: 0h base + 4h offset = 4h

| Bit   | 0   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|-------|-----|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| R     | ETV |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |     |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 1   | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1  | 0  | 0  | 1  | 0  | 1  | 0  | 1  | 0  | 0  | 0  | 1  | 1  | 0  | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 1  |

### M\_CAN\_ENDN field descriptions

| Field       | Description                                                       |
|-------------|-------------------------------------------------------------------|
| 0–31<br>ETV | Endianness Test Value<br>The endianness test value is 0x87654321. |

### 38.3.3 Data Bit Timing and Prescaler Register (M\_CAN\_DBTP)

This register is only writable if bits CCCR.CCE and CCCR.INIT are set. The CAN bit time may be programmed in the range of 4 to 49 time quanta. The CAN time quantum may be programmed in the range of 1 to 32 m\_can\_cclk periods.  $t_q = (DBRP + 1) \text{ mtq}$ .

DTSEG1 is the sum of Prop\_Seg and Phase\_Seg1. DTSEG2 is Phase\_Seg2.

Therefore the length of the bit time is (programmed values)  $[DTSEG1 + DTSEG2 + 3] t_q$  or (functional values)  $[Sync\_Seg + Prop\_Seg + Phase\_Seg1 + Phase\_Seg2] t_q$ .

The Information Processing Time (IPT) is zero, meaning the data for the next bit is available at the first clock edge after the sample point.

#### NOTE

With a M\_CAN clock of 8 MHz, the reset value of 0x00000A33 configures the M\_CAN for a data phase bit rate of 500 kBit/s.

#### NOTE

The bit rate configured for the CAN FD data phase via DBTP must be higher or equal to the bit rate configured for the arbitration phase via NBTP.

Address: 0h base + Ch offset = Ch

|       |   |   |   |   |   |   |   |   |     |   |    |      |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|-----|---|----|------|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8   | 9 | 10 | 11   | 12 | 13 | 14 | 15 |
| R     | 0 |   |   |   |   |   |   |   | TDC | 0 |    | DBRP |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |     |   |    |      |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0 | 0  | 0    | 0  | 0  | 0  | 0  |

|       |    |    |    |        |    |    |    |    |        |    |    |    |      |    |    |    |
|-------|----|----|----|--------|----|----|----|----|--------|----|----|----|------|----|----|----|
| Bit   | 16 | 17 | 18 | 19     | 20 | 21 | 22 | 23 | 24     | 25 | 26 | 27 | 28   | 29 | 30 | 31 |
| R     | 0  |    |    | DTSEG1 |    |    |    |    | DTSEG2 |    |    |    | DSJW |    |    |    |
| W     |    |    |    |        |    |    |    |    |        |    |    |    |      |    |    |    |
| Reset | 0  | 0  | 0  | 0      | 1  | 0  | 1  | 0  | 0      | 0  | 1  | 1  | 0    | 0  | 1  | 1  |

#### M\_CAN\_DBTP field descriptions

| Field           | Description                                                                                                                                                                            |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–7<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                |
| 8<br>TDC        | Transceiver Delay Compensation<br><br><b>NOTE:</b> This field has Protected Write status.<br><br>0 Transceiver Delay Compensation disabled<br>1 Transceiver Delay Compensation enabled |

Table continues on the next page...

**M\_CAN\_DBTP field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 9–10<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                           |
| 11–15<br>DBRP     | Data Bit Rate Prescaler<br><br><b>NOTE:</b> This field has Protected Write status.<br><br>0x0-0x1F The value by which the oscillator frequency is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid values for the Bit Rate Prescaler are 0 to 31. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used. |
| 16–18<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                           |
| 19–23<br>DTSEG1   | Data time segment before sample point<br><br><b>NOTE:</b> This field has Protected Write status.<br><br>0x0-0x1F Valid values are 0 to 31. The actual interpretation by the hardware of this value is such that one more than the programmed value is used.                                                                                                                                                                       |
| 24–27<br>DTSEG2   | Data time segment after sample point<br><br><b>NOTE:</b> This field has Protected Write status.<br><br>0x0-0xF Valid values are 0 to 15. The actual interpretation by the hardware of this value is such that one more than the programmed value is used.                                                                                                                                                                         |
| 28–31<br>DSJW     | Data (Re)Synchronization Jump Width<br><br><b>NOTE:</b> This field has Protected Write status.<br><br>0x0-0xF Valid values are 0 to 15. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.                                                                                                                                                                     |

**38.3.4 Test Register (M\_CAN\_TEST)**

Write access to the Test Register has to be enabled by setting bit CCCR[TEST] to 1. All Test Register functions are set to their reset values when bit CCCR[TEST] is reset.

Loop Back Mode and software control of pin M\_CAN\_Tx are hardware test modes. Programming of TX other than 00 may disturb the message transfer on the CAN bus.

Address: 0h base + 10h offset = 10h

|       |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |
|-------|----|----|----|----|----|----|----|----|----|----|----|------|----|----|----|----|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11   | 12 | 13 | 14 | 15 |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |
| W     |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0  | 0  | 0  | 0  |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27   | 28 | 29 | 30 | 31 |
| R     | 0  |    |    |    |    |    |    |    | RX | TX |    | LBCK | 0  |    |    |    |
| W     |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | u* | 0  | 0  | 0    | 0  | 0  | 0  | 0  |

\* Notes:

- u = Unaffected by reset.

### M\_CAN\_TEST field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                            |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–23<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                |
| 24<br>RX         | Receive Pin<br>Monitors the actual value of pin M_CAN Rx<br><br>0 The CAN bus is dominant (M_CAN Rx = 0)<br>1 The CAN bus is recessive (M_CAN Rx = 1)                                                                                                                                                                  |
| 25–26<br>TX      | Control of Transmit Pin<br><br><b>NOTE:</b> This field has Protected Write status.<br><br>00 Reset value, M_CAN Tx controlled by the CAN Core, updated at the end of the CAN bit time<br>01 Sample Point can be monitored at pin M_CAN Tx<br>10 Dominant (0) level at pin M_CAN Tx<br>11 Recessive (1) at pin M_CAN Tx |
| 27<br>LBCK       | Loopback mode<br><br><b>NOTE:</b> This field has Protected Write status.                                                                                                                                                                                                                                               |

Table continues on the next page...

**M\_CAN\_TEST field descriptions (continued)**

| Field             | Description                                                                             |
|-------------------|-----------------------------------------------------------------------------------------|
|                   | 0 Reset value, Loopback mode is disabled                                                |
|                   | 1 Loopback mode is enabled (see <a href="#">Test Modes</a> )                            |
| 28–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |

**38.3.5 RAM Watchdog Register (M\_CAN\_RWD)**

The RAM Watchdog monitors the READY output of the Message RAM. A Message RAM access via the M\_CAN's Generic Master Interface starts the Message RAM Watchdog Counter with the value configured by RWD[WDC]. The counter is reloaded with RWD[WDC] when the Message RAM signals successful completion by activating its READY output. In case there is no response from the Message RAM until the counter has counted down to zero, the counter stops and interrupt flag IR[WDI] is set. The RAM Watchdog Counter is clocked by the Host clock.

Address: 0h base + 14h offset = 14h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |     |    |    |    |    |    |    |    |     |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|-----|----|----|----|----|----|----|----|-----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16  | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24  | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | WDV |    |    |    |    |    |    |    | WDC |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |     |    |    |    |    |    |    |    |     |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**M\_CAN\_RWD field descriptions**

| Field            | Description                                                                                                                                                                           |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                               |
| 16–23<br>WDV     | Watchdog Value<br>Actual Message RAM Watchdog Counter Value.                                                                                                                          |
| 24–31<br>WDC     | Watchdog Configuration<br>Start value of the Message RAM Watchdog Counter. With the reset value of 00 the counter is disabled.<br><b>NOTE:</b> This field has Protected Write status. |

### 38.3.6 CC Control Register (M\_CAN\_CCCR)

For details about setting and resetting of single bits see [Software Initialization](#).

Address: 0h base + 18h offset = 18h

|       |      |     |      |      |    |    |      |      |      |     |     |     |     |     |     |      |
|-------|------|-----|------|------|----|----|------|------|------|-----|-----|-----|-----|-----|-----|------|
| Bit   | 0    | 1   | 2    | 3    | 4  | 5  | 6    | 7    | 8    | 9   | 10  | 11  | 12  | 13  | 14  | 15   |
| R     | 0    |     |      |      |    |    |      |      |      |     |     |     |     |     |     |      |
| W     |      |     |      |      |    |    |      |      |      |     |     |     |     |     |     |      |
| Reset | 0    | 0   | 0    | 0    | 0  | 0  | 0    | 0    | 0    | 0   | 0   | 0   | 0   | 0   | 0   | 0    |
| Bit   | 16   | 17  | 18   | 19   | 20 | 21 | 22   | 23   | 24   | 25  | 26  | 27  | 28  | 29  | 30  | 31   |
| R     |      |     |      |      | 0  |    |      |      |      |     |     |     | CSA |     |     |      |
| W     | NISO | TXP | EFBI | PHDX |    |    | BRSE | FDOE | TEST | DAR | MON | CSR |     | ASM | CCE | INIT |
| Reset | 0    | 0   | 0    | 0    | 0  | 0  | 0    | 0    | 0    | 0   | 0   | 0   | 0   | 0   | 0   | 1    |

**M\_CAN\_CCCR field descriptions**

| Field            | Description                                                                                                                                                                                                                                                                                                                    |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                        |
| 16<br>NISO       | Non ISO Operation<br><br>If this bit is set, the M_CAN uses the CAN FD frame format as specified by the Bosch CAN FD Specification V1.0.<br><br><b>NOTE:</b> This field has Protected Write status.<br><br>0 CAN FD frame format according to ISO11898-1<br>1 CAN FD frame format according to Bosch CAN FD Specification V1.0 |
| 17<br>TXP        | Transmit Pause<br><br>If this bit is set, the M_CAN pauses for two CAN bit times before starting the next transmission after itself has successfully transmitted a frame (see <a href="#">Tx Handling</a> ).<br><br><b>NOTE:</b> This field has Protected Write status.                                                        |

*Table continues on the next page...*

**M\_CAN\_CCCR field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                       |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | 0 Transmit pause disabled<br>1 Transmit pause enabled                                                                                                                                                                                                                                                                             |
| 18<br>EFBI        | Edge Filtering during Bus Integration<br><br><b>NOTE:</b> This field has Protected Write status.<br><br>0 Edge filtering disabled<br>1 Two consecutive dominant tq required to detect an edge for hard synchronization                                                                                                            |
| 19<br>PHDX        | Protocol Exception Handling Disable<br><br><b>NOTE:</b> When protocol exception handling is disabled, the M_CAN will transmit an error frame when it detects a protocol exception condition.<br><br>This field has Protected Write status.<br><br>0 Protocol exception handling enabled<br>1 Protocol exception handling disabled |
| 20–21<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                           |
| 22<br>BRSE        | Bit Rate Switch Enable<br><br><b>NOTE:</b> When CAN FD operation is disabled FDOE = 0, BRSE is not evaluated.<br><br>This field has Protected Write status.<br><br>0 Bit rate switching for transmissions disabled<br>1 Bit rate switching for transmissions enabled                                                              |
| 23<br>FDOE        | FD Operation Enable<br><br><b>NOTE:</b> This field has Protected Write status.<br><br>0 FD operation disabled<br>1 FD operation enabled                                                                                                                                                                                           |
| 24<br>TEST        | Test Mode Enable<br><br><b>NOTE:</b> This field has Protected Write status.<br><br>0 Normal operation, register TEST holds reset values<br>1 Test Mode, write access to register TEST enabled                                                                                                                                     |
| 25<br>DAR         | Disable Automatic Retransmission<br><br><b>NOTE:</b> This field has Protected Write status.<br><br>0 Automatic retransmission of messages not transmitted successfully enabled<br>1 Automatic retransmission disabled                                                                                                             |
| 26<br>MON         | Bus Monitoring Mode                                                                                                                                                                                                                                                                                                               |

*Table continues on the next page...*



**M\_CAN\_CCCR field descriptions (continued)**

| Field      | Description                                                                                                                                                                                                                                                                                                                                                                                         |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|            | <p>Bit MON can only be set by the Host when both CCE and INIT are set to '1'. The bit can be reset by the Host at any time.</p> <p><b>NOTE:</b> This field has Protected Write status.</p> <p>0 Bus Monitoring Mode is disabled<br/>1 Bus Monitoring Mode is enabled</p>                                                                                                                            |
| 27<br>CSR  | <p>Clock Stop Request</p> <p>0 No clock stop is requested<br/>1 Clock stop requested. When clock stop is requested, first INIT and then CSA will be set after all pending transfer requests have been completed and the CAN bus reached idle.</p>                                                                                                                                                   |
| 28<br>CSA  | <p>Clock Stop Acknowledge</p> <p>0 No clock stop acknowledged<br/>1 M_CAN may be set in power down by stopping M_CAN input clocks.</p>                                                                                                                                                                                                                                                              |
| 29<br>ASM  | <p>Restricted Operation Mode</p> <p>Bit ASM can only be set by the Host when both CCE and INIT are set to '1'. The bit can be reset by the Host at any time. For a description of the Restricted Operation Mode see <a href="#">Restricted Operation Mode</a>.</p> <p>0 Normal CAN operation<br/>1 Restricted Operation Mode active</p>                                                             |
| 30<br>CCE  | <p>Configuration Change Enable</p> <p><b>NOTE:</b> This field has Protected Write status.</p> <p>0 The CPU has no write access to the protected configuration registers.<br/>1 The CPU has write access to the protected configuration registers (while CCCR[INIT] = 1)</p>                                                                                                                         |
| 31<br>INIT | <p>Initialization</p> <p><b>NOTE:</b> Due to the synchronization mechanism between the two clock domains, there may be a delay until the value written to INIT can be read back. Therefore the programmer has to assure that the previous value written to INIT has been accepted by reading INIT before setting INIT to a new value.</p> <p>0 Normal Operation<br/>1 Initialization is started</p> |

**38.3.7 Nominal Bit Timing and Prescaler Register (M\_CAN\_NBTP)**

This register is only writable if its CCCR[CCE] and CCCR[INIT] are set. The CAN bit time may be programmed in the range of 4 to 385 time quanta. The CAN time quantum may be programmed in the range of 1 to 512 m\_can\_cclk periods.  $tq = (NBRP + 1) mtq$ .

NTSEG1 is the sum of Prop\_Seg and Phase\_Seg1. NTSEG2 is Phase\_Seg2.

## Memory Map and Register Description

Therefore the length of the bit time is (programmed values)  $[NTSEG1 + NTSEG2 + 3] t_q$  or (functional values)  $[Sync\_Seg + Prop\_Seg + Phase\_Seg1 + Phase\_Seg2] t_q$ .

The Information Processing Time (IPT) is zero, meaning the data for the next bit is available at the first clock edge after the sample point.

### NOTE

With a CAN clock of 8 MHz, the reset value of 0x06000A03 configures the M\_CAN for a bit rate of 500 kBit/s.

Address: 0h base + 1Ch offset = 1Ch

|       |        |    |    |    |    |    |    |    |      |        |    |    |    |    |    |    |
|-------|--------|----|----|----|----|----|----|----|------|--------|----|----|----|----|----|----|
| Bit   | 0      | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8    | 9      | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | NSJW   |    |    |    |    |    |    |    | NBRP |        |    |    |    |    |    |    |
| W     |        |    |    |    |    |    |    |    |      |        |    |    |    |    |    |    |
| Reset | 0      | 0  | 0  | 0  | 0  | 1  | 1  | 0  | 0    | 0      | 0  | 0  | 0  | 0  | 0  | 0  |
| Bit   | 16     | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24   | 25     | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | NTSEG1 |    |    |    |    |    |    |    | 0    | NTSEG2 |    |    |    |    |    |    |
| W     |        |    |    |    |    |    |    |    |      |        |    |    |    |    |    |    |
| Reset | 0      | 0  | 0  | 0  | 1  | 0  | 1  | 0  | 0    | 0      | 0  | 0  | 0  | 0  | 1  | 1  |

### M\_CAN\_NBTP field descriptions

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–6<br>NSJW     | Nominal (Re) Synchronization Jump Width<br><br><b>NOTE:</b> This field has Protected Write status.<br><br>0x00-0x7F Valid values are 0 to 127. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.                                                                                                                                                                      |
| 7–15<br>NBRP    | Nominal Baud Rate Prescaler<br><br><b>NOTE:</b> This field has Protected Write status.<br><br>0x000-0x1FF The value by which the oscillator frequency is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid values for the Bit Rate Prescaler are 0 to 511. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used. |
| 16–23<br>NTSEG1 | Nominal Time segment before sample point<br><br><b>NOTE:</b> This field has Protected Write status.<br><br>0x01-0xFF Valid values are 1 to 255. The actual interpretation by the hardware of this value is such that one more than the programmed value is used.                                                                                                                                                                          |
| 24<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                   |
| 25–31<br>NTSEG2 | Nominal Time segment after sample point<br><br><b>NOTE:</b> This field has Protected Write status.                                                                                                                                                                                                                                                                                                                                        |

Table continues on the next page...

**M\_CAN\_NBTP field descriptions (continued)**

| Field     | Description                                                                                                                                      |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x00-0x7F | Valid values are 0 to 127. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used. |

**38.3.8 Timestamp Counter Configuration Register (M\_CAN\_TSCC)**

For description of the Timestamp Counter see [Timestamp Generation](#)

Address: 0h base + 20h offset = 20h

|       |   |   |   |   |   |   |   |   |   |   |    |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11  | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29  | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    | TCP |    |    |    | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    | TSS |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0  |    |

**M\_CAN\_TSCC field descriptions**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                        |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–11<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                            |
| 12–15<br>TCP      | Timestamp Counter Prescaler<br><br><b>NOTE:</b> This field has Protected Write status.<br><b>NOTE:</b> With CAN FD, timestamp generation is not supported.<br><br>0x0-0xF Configures the timestamp and timeout counters time unit in multiples of CAN bit times [1...16].<br>The actual interpretation by the hardware of this value is such that one more than the value programmed here is used. |
| 16–29<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                            |
| 30–31<br>TSS      | Timestamp Select<br><br><b>NOTE:</b> This field has Protected Write status.<br><br>00 Timestamp counter value always 0x0000<br>01 Timestamp counter value incremented according to TCP<br>10 Reserved<br>11 Same as 00                                                                                                                                                                             |

### 38.3.9 Timestamp Counter Value Register (M\_CAN\_TSCV)

Address: 0h base + 24h offset = 24h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16  | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | TSC |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | w1c |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

#### M\_CAN\_TSCV field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                  |
| 16–31<br>TSC     | Timestamp Counter<br><br>The internal Timestamp Counter value is captured on start of frame (both Rx and Tx). When TSCC[TSS] = 01, the Timestamp Counter is incremented in multiples of CAN bit times [1...16] depending on the configuration of TSCC[TCP]. A wrap around sets interrupt flag IR[TSW]. Write access resets the counter to zero.<br><br><b>NOTE:</b> A "wrap around" is a change of the Timestamp Counter value from non-zero to zero not caused by write access to TSCV. |

### 38.3.10 Timeout Counter Configuration Register (M\_CAN\_TOCC)

See [Timeout Counter](#) for description of the Timeout Counter or for use of timeout function with CAN FD.

Address: 0h base + 28h offset = 28h

|       |     |    |    |    |    |    |    |    |    |    |    |    |     |    |      |    |
|-------|-----|----|----|----|----|----|----|----|----|----|----|----|-----|----|------|----|
| Bit   | 0   | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12  | 13 | 14   | 15 |
| R     | TOP |    |    |    |    |    |    |    |    |    |    |    |     |    |      |    |
| W     |     |    |    |    |    |    |    |    |    |    |    |    |     |    |      |    |
| Reset | 1   | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1   | 1  | 1    | 1  |
| Bit   | 16  | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28  | 29 | 30   | 31 |
| R     | 0   |    |    |    |    |    |    |    |    |    |    |    | TOS |    | ETOC |    |
| W     |     |    |    |    |    |    |    |    |    |    |    |    |     |    |      |    |
| Reset | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0  | 0    | 0  |

**M\_CAN\_TOCC field descriptions**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>TOP       | Timeout Period<br>Start value of the Timeout Counter (down-counter). Configures the Timeout Period.<br><b>NOTE:</b> This field has Protected Write status.                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 16–28<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 29–30<br>TOS      | Timeout Select<br>When operating in Continuous mode, a write to TOCV presets the counter to the value configured by TOCC[TOP] and continues down-counting. When the Timeout Counter is controlled by one of the FIFOs, an empty FIFO presets the counter to the value configured by TOCC[TOP]. Down-counting is started when the first FIFO element is stored.<br><b>NOTE:</b> This field has Protected Write status.<br><br>00 Continuous operation<br>01 Timeout controlled by Tx Event FIFO<br>10 Timeout controlled by Rx FIFO 0<br>11 Timeout controlled by Rx FIFO 1 |
| 31<br>ETOC        | Enable Timeout Counte<br><br><b>NOTE:</b> This field has Protected Write status.<br><br>0 Timeout Counter disabled<br>1 Timeout Counter enabled                                                                                                                                                                                                                                                                                                                                                                                                                            |

**38.3.11 Timeout Counter Value Register (M\_CAN\_TOCV)**

Address: 0h base + 2Ch offset = 2Ch

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16  | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | TOC |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | w1c |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 1   | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  |    |

**M\_CAN\_TOCV field descriptions**

| Field            | Description                                                                                                                                                                                                                                                                                           |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                               |
| 16–31<br>TOC     | Timeout Counter<br>The Timeout Counter is decremented in multiples of CAN bit times [1...16] depending on the configuration of TSCC[TCP]. When decremented to zero, interrupt flag IR[TOO] is set and the Timeout Counter is stopped. Start and reset/restart conditions are configured via TOCC[TOS] |

### 38.3.12 Error Counter Register (M\_CAN\_ECR)

#### NOTE

When CCCR[ASM] is set, the CAN protocol controller does not increment TEC and REC when a CAN protocol error is detected, but CEL is still incremented.

Address: 0h base + 40h offset = 40h

|       |    |     |    |    |    |    |    |    |     |    |    |    |    |    |    |    |  |
|-------|----|-----|----|----|----|----|----|----|-----|----|----|----|----|----|----|----|--|
| Bit   | 0  | 1   | 2  | 3  | 4  | 5  | 6  | 7  | 8   | 9  | 10 | 11 | 12 | 13 | 14 | 15 |  |
| R     | 0  |     |    |    |    |    |    |    | CEL |    |    |    |    |    |    |    |  |
| W     |    |     |    |    |    |    |    |    | CEL |    |    |    |    |    |    |    |  |
| Reset | 0  | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  |  |
| Bit   | 16 | 17  | 18 | 19 | 20 | 21 | 22 | 23 | 24  | 25 | 26 | 27 | 28 | 29 | 30 | 31 |  |
| R     | RP | REC |    |    |    |    |    |    | TEC |    |    |    |    |    |    |    |  |
| W     |    |     |    |    |    |    |    |    |     |    |    |    |    |    |    |    |  |
| Reset | 0  | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  |  |

#### M\_CAN\_ECR field descriptions

| Field           | Description                                                                                                                                                                                                                                                                                       |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–7<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                           |
| 8–15<br>CEL     | CAN Error Logging<br>The counter is incremented each time when a CAN protocol error causes the Transmit Error Counter or the Receive Error Counter to be incremented. It is reset by read access to CEL. The counter stops at 0xFF; the next increment of TEC or REC sets interrupt flag IR[ELO]. |
| 16<br>RP        | Receive Error Passive<br>0 The Receive Error Counter is below the error passive level of 128<br>1 The Receive Error Counter has reached the error passive level of 128                                                                                                                            |
| 17–23<br>REC    | Receive Error Counter<br>Actual state of the Receive Error Counter, values between 0 and 127.                                                                                                                                                                                                     |
| 24–31<br>TEC    | Transmit Error Counter<br>Actual state of the Transmit Error Counter, values between 0 and 255.                                                                                                                                                                                                   |

### 38.3.13 Protocol Status Register (M\_CAN\_PSR)

#### NOTE

When a frame in CAN FD format has reached the data phase with BRS flag set, the next CAN event (error or valid frame) will be shown in DLEC instead of LEC. An error in a fixed stuff bit of a CAN FD CRC sequence will be shown as a Form Error, not Stuff Error.

#### NOTE

The Bus\_Off recovery sequence (see CAN Specification Rev. 2.0 or ISO11898-1) cannot be shortened by setting or resetting CCCR[INIT].

1. If the device goes Bus\_Off, it sets CCCR[INIT] and PSR[BO], stopping all bus activities.
2. After the user clears CCCR[INIT], the user must then wait for 129 occurrences of Bus Idle (129 \* 11 consecutive recessive bits) before normal CAN operation can resume.
  - During the waiting period, each time the device monitors a sequence of 11 recessive bits, a Bit0Error code (101b) is written to PSR[LEC].
  - These code writes enable the user to check readily whether the CAN bus is stuck at dominant or continuously disturbed and to monitor the Bus\_Off recovery sequence.
  - ECR[REC] is used to count these sequences.
3. After the observation of the 129 occurrences of Bus Idle, the device clears PSR[BO].

Address: 0h base + 44h offset = 44h

|       |   |   |   |   |   |   |   |   |      |   |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|------|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8    | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0 |   |   |   |   |   |   |   | TDCV |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |      |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0    | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |     |      |      |      |      |    |    |    |    |    |     |    |     |    |    |
|-------|----|-----|------|------|------|------|----|----|----|----|----|-----|----|-----|----|----|
| Bit   | 16 | 17  | 18   | 19   | 20   | 21   | 22 | 23 | 24 | 25 | 26 | 27  | 28 | 29  | 30 | 31 |
| R     | 0  | PXE | RFDF | RBRS | RESI | DLEC |    |    | BO | EW | EP | ACT |    | LEC |    |    |
| W     |    |     |      |      |      |      |    |    |    |    |    |     |    |     |    |    |
| Reset | 0  | 0   | 0    | 0    | 0    | 1    | 1  | 1  | 0  | 0  | 0  | 0   | 0  | 1   | 1  | 1  |

## M\_CAN\_PSR field descriptions

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                        |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–8<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                            |
| 9–15<br>TDCV    | Transmitter Delay Compensation Value<br><br>0x00-0x7F Position of the secondary sample point, defined by the sum of the measured delay from M_CAN Tx to M_CAN Rx and TDCR[TDCO]. The SSP position is, in the data phase, the number of mtq between the start of the transmitted bit and the secondary sample point. Valid values are 0 to 127 mtq.                                 |
| 16<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                            |
| 17<br>PXE       | Protocol Exception Event<br><br><b>NOTE:</b> This field is reset by a read operation.<br><br>0 No protocol exception event occurred since last read access<br>1 Protocol exception event occurred                                                                                                                                                                                  |
| 18<br>RFDF      | Received a CAN FD Message<br><br>This bit is set independent of acceptance filtering.<br><br><b>NOTE:</b> This field is reset by a read operation.<br><br>0 Since this bit was reset by the CPU, no CAN FD message has been received<br>1 Message in CAN FD format with FDF flag set has been received                                                                             |
| 19<br>RBRS      | BRS flag of last received CAN FD Message<br><br>This bit is set together with RFDF, independent of acceptance filtering.<br><br><b>NOTE:</b> This field is reset by a read operation.<br><br>0 Last received CAN FD message did not have its BRS flag set<br>1 Last received CAN FD message had its BRS flag set                                                                   |
| 20<br>RESI      | ESI flag of last received CAN FD Message<br><br>This bit is set together with RFDF, independent of acceptance filtering.<br><br><b>NOTE:</b> This field is reset by a read operation.<br><br>0 Last received CAN FD message did not have its ESI flag set<br>1 Last received CAN FD message had its ESI flag set                                                                   |
| 21–23<br>DLEC   | Data Phase Last Error Code<br><br>Type of last error that occurred in the data phase of a CAN FD format frame with its BRS flag set. Coding is the same as for LEC. This field will be cleared to zero when a CAN FD format frame with its BRS flag set has been transferred (reception or transmission) without error.<br><br><b>NOTE:</b> This field is set by a read operation. |
| 24<br>BO        | Bus_Off Status<br><br>0 The M_CAN is not Bus_Off<br>1 The M_CAN is in Bus_Off state                                                                                                                                                                                                                                                                                                |

Table continues on the next page...



## M\_CAN\_PSR field descriptions (continued)

| Field        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 25<br>EW     | Warning Status<br>0 Both error counters are below the Error_Warning limit of 96<br>1 At least one of error counter has reached the Error_Warning limit of 96                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 26<br>EP     | Error Passive<br>0 The M_CAN is in the Error_Active state. It normally takes part in bus communication and sends an active error flag when an error has been detected<br>1 The M_CAN is in the Error_Passive state                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 27–28<br>ACT | Activity<br>Monitors the module's CAN communication state.<br><b>NOTE:</b> ACT is set to 00 by a Protocol Exception Event.<br>00 Synchronizing - node is synchronizing on CAN communication<br>01 Idle - node is neither receiver nor transmitter<br>10 Receiver - node is operating as receiver<br>11 Transmitter - node is operating as transmitter                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 29–31<br>LEC | Last Error Code<br>The LEC indicates the type of the last error to occur on the CAN bus. This field will be cleared to 0 when a message has been transferred (reception or transmission) without error.<br><b>NOTE:</b> This field is set by a read operation.<br>000 Error: No error occurred since LEC has been reset by successful reception or transmission<br>001 Stuff Error: More than 5 equal bits in a sequence have occurred in a part of a received message where this is not allowed<br>010 Form Error: A fixed format part of a received frame has the wrong format<br>011 AckError: The message transmitted by the M_CAN was not acknowledged by another node.<br>100 Bit1Error: During the transmission of a message (with the exception of the arbitration field), the device wanted to send a recessive level (bit of logical value 1), but the monitored bus value was dominant.<br>101 Bit0Error: During the transmission of a message (or acknowledge bit, or active error flag, or overload flag), the device wanted to send a dominant level (data or identifier bit logical value '0'), but the monitored bus value was recessive. During Bus_Off recovery this status is set each time a sequence of 11 recessive bits has been monitored. This enables the CPU to monitor the proceeding of the Bus_Off recovery sequence (indicating the bus is not stuck at dominant or continuously disturbed).<br>110 CRCError: The CRC check sum of a received message was incorrect. The CRC of an incoming message does not match with the CRC calculated from the received data.<br>111 NoChange: Any read access to the Protocol Status Register re-initializes the LEC to 7. When the LEC shows the value 7, no CAN bus event was detected since the last CPU read access to the Protocol Status Register. |

### 38.3.14 Transmitter Delay Compensation Register (M\_CAN\_TDCR)

Address: 0h base + 48h offset = 48h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |      |    |    |    |    |    |    |    |    |      |    |    |    |    |    |  |
|-------|----|------|----|----|----|----|----|----|----|----|------|----|----|----|----|----|--|
| Bit   | 16 | 17   | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26   | 27 | 28 | 29 | 30 | 31 |  |
| R     | 0  | TDCO |    |    |    |    |    |    |    | 0  | TDCF |    |    |    |    |    |  |
| W     |    |      |    |    |    |    |    |    |    |    |      |    |    |    |    |    |  |
| Reset | 0  | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0  | 0  | 0  | 0  | 0  |  |

#### M\_CAN\_TDCR field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                           |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–16<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                               |
| 17–23<br>TDCO    | Transmitter Delay Compensation Offset.<br><br><b>NOTE</b><br>This field has Protected Write status.<br><br>0x00-0x7F Offset value defining the distance between the measured delay from M_CAN Tx to M_CAN Rx and the secondary sample point. Valid values are 0 to 127 mtq.                                                                                                                                           |
| 24<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                               |
| 25–31<br>TDCF    | Transmitter Delay Compensation Filter Window Length.<br><br><b>NOTE</b><br>This field has Protected Write status.<br><br>0x00-0x7F Defines the minimum value for the SSP position, dominant edges on M_CAN Rx that would result in an earlier SSP position are ignored for transmitter delay measurement. The feature is enabled when TDCF is configured to a value greater than TDCO. Valid values are 0 to 127 mtq. |

### 38.3.15 Interrupt Register (M\_CAN\_IR)

The flags are set when one of the listed conditions is detected (edge-sensitive). The flags remain set until the Host clears them. Aflag is cleared by writing a 1 to the corresponding bit position. Writing a 0 has no effect. Ahard reset will clear the register. The configuration of IE controls whether an interrupt is generated. The configuration of ILS controls on which interrupt line an interrupt is signalled.

Address: 0h base + 50h offset = 50h

|       |   |   |     |     |     |     |    |    |    |     |     |     |     |     |      |     |
|-------|---|---|-----|-----|-----|-----|----|----|----|-----|-----|-----|-----|-----|------|-----|
| Bit   | 0 | 1 | 2   | 3   | 4   | 5   | 6  | 7  | 8  | 9   | 10  | 11  | 12  | 13  | 14   | 15  |
| R     | 0 |   |     |     |     |     |    |    |    |     |     |     |     |     |      |     |
| W     |   |   | ARA | PED | PEA | WDI | BO | EW | EP | ELO | BEU | BEC | DRX | TOO | MRAF | TSW |
| Reset | 0 | 0 | 0   | 0   | 0   | 0   | 0  | 0  | 0  | 0   | 0   | 0   | 0   | 0   | 0    | 0   |

|       |      |      |      |      |     |     |    |     |      |      |      |      |      |      |      |      |
|-------|------|------|------|------|-----|-----|----|-----|------|------|------|------|------|------|------|------|
| Bit   | 16   | 17   | 18   | 19   | 20  | 21  | 22 | 23  | 24   | 25   | 26   | 27   | 28   | 29   | 30   | 31   |
| R     |      |      |      |      |     |     |    |     |      |      |      |      |      |      |      |      |
| W     | TEFL | TEFF | TEFW | TEFN | TFE | TCF | TC | HPM | RF1L | RF1F | RF1W | RF1N | RF0L | RF0F | RF0W | RF0N |
| Reset | 0    | 0    | 0    | 0    | 0   | 0   | 0  | 0   | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |

**M\_CAN\_IR field descriptions**

| Field           | Description                                                                                                                                                   |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–1<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                       |
| 2<br>ARA        | Access to Reserved Address<br>0 No access to reserved address occurred<br>1 Access to reserved address occurred                                               |
| 3<br>PED        | Protocol Error in Data Phase<br>0 No protocol error in data phase<br>1 Protocol error in data phase detected (PSR[DLEC] not equal to 0,7)                     |
| 4<br>PEA        | Protocol Error in Arbitration Phase<br>0 No protocol error in arbitration phase<br>1 Protocol error in arbitration phase detected (PSR[LEC] not equal to 0,7) |
| 5<br>WDI        | Watchdog Interrupt<br>0 No Message RAM Watchdog event occurred<br>1 Message RAM Watchdog event due to missing READY                                           |
| 6<br>BO         | Bus_Off Status                                                                                                                                                |

*Table continues on the next page...*

**M\_CAN\_IR field descriptions (continued)**

| Field      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|            | 0 Bus_Off status unchanged<br>1 Bus_Off status changed                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 7<br>EW    | Warning Status<br>0 Error_Warning status unchanged<br>1 Error_Warning status changed                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 8<br>EP    | Error Passive<br>0 Error_Passive status unchanged<br>1 Error_Passive status changed                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 9<br>ELO   | Error Logging Overflow<br>0 CAN Error Logging Counter did not overflow<br>1 Overflow of CAN Error Logging Counter occurred                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 10<br>BEU  | Bit Error Uncorrected<br>Message RAM bit error detected, uncorrected. Controlled by an external ECC logic attached to the Message RAM. An uncorrected Message RAM bit error sets CCCR[INIT] to 1. This is done to avoid transmission of corrupted data.<br>0 No bit error detected when reading from Message RAM<br>1 Bit error detected, uncorrected (e.g. parity logic)                                                                                                                                                                                                                                                                                                                                             |
| 11<br>BEC  | Bit Error Corrected<br>Message RAM bit error detected and corrected. Controlled by an external parity ECC logic attached to the Message RAM.<br>0 No bit error detected when reading from Message RAM<br>1 Bit error detected and corrected (e.g. ECC)                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 12<br>DRX  | Message stored to Dedicated Rx Buffer<br>The flag is set whenever a received message has been stored into a dedicated Rx Buffer.<br>0 No Rx Buffer updated<br>1 At least one received message stored into a Rx Buffer                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 13<br>TOO  | Timeout Occurred<br>0 No timeout<br>1 Timeout reached                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 14<br>MRAF | Message RAM Access Failure<br>The flag is set, when the Rx Handler <ul style="list-style-type: none"> <li>• has not completed acceptance filtering or storage of an accepted message until the arbitration field of the following message has been received. In this case acceptance filtering or message storage is aborted and the Rx Handler starts processing of the following message.</li> <li>• was not able to write a message to the Message RAM. In this case message storage is aborted.</li> </ul> In both cases the FIFO put index is not updated resp. the New Data flag for a dedicated Rx Buffer is not set, a partly stored message is overwritten when the next message is stored to this location. |

*Table continues on the next page...*

**M\_CAN\_IR field descriptions (continued)**

| Field      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|            | <p>The flag is also set when the Tx Handler was not able to read a message from the Message RAM in time. In this case message transmission is aborted. In case of a Tx Handler access failure the M_CAN is switched into Restricted Operation Mode (see <a href="#">Restricted Operation Mode</a>). To leave Restricted Operation Mode, the CPU has to reset CCCR[ASM].</p> <p>0 No Message RAM access failure occurred<br/>1 Message RAM access failure occurred</p> |
| 15<br>TSW  | <p>Timestamp Wraparound</p> <p>0 No timestamp counter wrap-around<br/>1 Timestamp counter wrapped around</p>                                                                                                                                                                                                                                                                                                                                                          |
| 16<br>TEFL | <p>Tx Event FIFO Element Lost</p> <p>0 No Tx Event FIFO element lost<br/>1 Tx Event FIFO element lost, also set after write attempt to Tx Event FIFO of size zero</p>                                                                                                                                                                                                                                                                                                 |
| 17<br>TEFF | <p>Tx Event FIFO Full</p> <p>0 Tx Event FIFO not full<br/>1 Tx Event FIFO full</p>                                                                                                                                                                                                                                                                                                                                                                                    |
| 18<br>TEFW | <p>Tx Event FIFO Watermark Reached</p> <p>0 Tx Event FIFO fill level below watermark<br/>1 Tx Event FIFO fill level reached watermark</p>                                                                                                                                                                                                                                                                                                                             |
| 19<br>TEFN | <p>Tx Event FIFO New Entry</p> <p>0 Tx Event FIFO unchanged<br/>1 Tx Handler wrote Tx Event FIFO element</p>                                                                                                                                                                                                                                                                                                                                                          |
| 20<br>TFE  | <p>Tx FIFO Empty</p> <p>0 Tx FIFO non-empty<br/>1 Tx FIFO empty</p>                                                                                                                                                                                                                                                                                                                                                                                                   |
| 21<br>TCF  | <p>Transmission Cancellation Finished</p> <p>0 No transmission cancellation finished<br/>1 Transmission cancellation finished</p>                                                                                                                                                                                                                                                                                                                                     |
| 22<br>TC   | <p>Transmission Completed</p> <p>0 Transmission not completed<br/>1 Transmission completed</p>                                                                                                                                                                                                                                                                                                                                                                        |
| 23<br>HPM  | <p>High Priority Message</p> <p>0 No high priority message received<br/>1 High priority message received</p>                                                                                                                                                                                                                                                                                                                                                          |
| 24<br>RF1L | <p>Rx FIFO 1 Message Lost</p> <p>0 No Rx FIFO 1 message lost<br/>1 Rx FIFO 1 message lost, also set after write attempt to Rx FIFO 1 of size zero</p>                                                                                                                                                                                                                                                                                                                 |
| 25<br>RF1F | <p>Rx FIFO 1 Full</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

*Table continues on the next page...*

**M\_CAN\_IR field descriptions (continued)**

| Field      | Description                                                                                                                                   |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
|            | 0 Rx FIFO 1 not full<br>1 Rx FIFO 1 full                                                                                                      |
| 26<br>RF1W | Rx FIFO 1 Watermark Reached<br><br>0 Rx FIFO 1 fill level below watermark<br>1 Rx FIFO 1 fill level reached watermark                         |
| 27<br>RF1N | Rx FIFO 1 New Message<br><br>0 No new message written to Rx FIFO 1<br>1 New message written to Rx FIFO 1                                      |
| 28<br>RF0L | Rx FIFO 0 Message Lost<br><br>0 No Rx FIFO 0 message lost<br>1 Rx FIFO 0 message lost, also set after write attempt to Rx FIFO 0 of size zero |
| 29<br>RF0F | Rx FIFO 0 Full<br><br>0 Rx FIFO 0 not full<br>1 Rx FIFO 0 full                                                                                |
| 30<br>RF0W | Rx FIFO 0 Watermark Reached<br><br>0 Rx FIFO 0 fill level below watermark<br>1 Rx FIFO 0 fill level reached watermark                         |
| 31<br>RF0N | Rx FIFO 0 New Message<br><br>0 No new message written to Rx FIFO 0<br>1 New message written to Rx FIFO 0                                      |

**38.3.16 Interrupt Enable Register (M\_CAN\_IE)**

The settings in the Interrupt Enable register determine which status changes in the Interrupt Register will be signaled on an interrupt line.

Address: 0h base + 54h offset = 54h

|       |       |       |       |       |      |      |     |      |       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|------|------|-----|------|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit   | 0     | 1     | 2     | 3     | 4    | 5    | 6   | 7    | 8     | 9     | 10    | 11    | 12    | 13    | 14    | 15    |
| R     | 0     |       | ARAE  | PEDE  | PEAE | WDIE | BOE | EWE  | EPE   | ELOE  | BEUE  | BECE  | DRXE  | TOOE  | MRAFE | TSWE  |
| W     |       |       |       |       |      |      |     |      |       |       |       |       |       |       |       |       |
| Reset | 0     | 0     | 0     | 0     | 0    | 0    | 0   | 0    | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |
| Bit   | 16    | 17    | 18    | 19    | 20   | 21   | 22  | 23   | 24    | 25    | 26    | 27    | 28    | 29    | 30    | 31    |
| R     | TEFLE | TEFFE | TEFWE | TEFNE | TFEE | TCFE | TCE | HPME | RF1LE | RF1FE | RF1WE | RF1NE | RF0LE | RF0FE | RF0WE | RF0NE |
| W     |       |       |       |       |      |      |     |      |       |       |       |       |       |       |       |       |
| Reset | 0     | 0     | 0     | 0     | 0    | 0    | 0   | 0    | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

**M\_CAN\_IE field descriptions**

| Field           | Description                                                                                           |
|-----------------|-------------------------------------------------------------------------------------------------------|
| 0–1<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.               |
| 2<br>ARAE       | Access to Reserved Address Enable                                                                     |
| 3<br>PEDE       | Protocol Error in Data Phase Enable                                                                   |
| 4<br>PEAE       | Protocol Error in Arbitration Phase Enable                                                            |
| 5<br>WDIE       | Watchdog Interrupt Enable<br>0 Interrupt disabled<br>1 Interrupt enabled                              |
| 6<br>BOE        | Bus_Off Status Interrupt Enable<br>0 Interrupt disabled<br>1 Interrupt enabled                        |
| 7<br>EWE        | Warning Status Interrupt Enable<br>0 Interrupt disabled<br>1 Interrupt enabled                        |
| 8<br>EPE        | Error Passive Interrupt Enable<br>0 Interrupt disabled<br>1 Interrupt enabled                         |
| 9<br>ELOE       | Error Logging Overflow Interrupt Enable<br>0 Interrupt disabled<br>1 Interrupt enabled                |
| 10<br>BEUE      | Bit Error Uncorrected Interrupt Enable<br>0 Interrupt disabled<br>1 Interrupt enabled                 |
| 11<br>BECE      | Bit Error Corrected Interrupt Enable<br>0 Interrupt disabled<br>1 Interrupt enabled                   |
| 12<br>DRXE      | Message stored to Dedicated Rx Buffer Interrupt Enable<br>0 Interrupt disabled<br>1 Interrupt enabled |
| 13<br>TOOE      | Timeout Occurred Interrupt Enable<br>0 Interrupt disabled<br>1 Interrupt enabled                      |
| 14<br>MRAFE     | Message RAM Access Failure Interrupt Enable<br>0 Interrupt disabled<br>1 Interrupt enabled            |

*Table continues on the next page...*

**M\_CAN\_IE field descriptions (continued)**

| Field       | Description                                                                                        |
|-------------|----------------------------------------------------------------------------------------------------|
| 15<br>TSWE  | Timestamp Wraparound Interrupt Enable<br>0 Interrupt disabled<br>1 Interrupt enabled               |
| 16<br>TEFLE | Tx Event FIFO Element Lost Interrupt Enable<br>0 Interrupt disabled<br>1 Interrupt enabled         |
| 17<br>TEFFE | Tx Event FIFO Full Interrupt Enable<br>0 Interrupt disabled<br>1 Interrupt enabled                 |
| 18<br>TEFWE | Tx Event FIFO Watermark Reached Interrupt Enable<br>0 Interrupt disabled<br>1 Interrupt enabled    |
| 19<br>TEFNE | Tx Event FIFO New Entry Interrupt Enable<br>0 Interrupt disabled<br>1 Interrupt enabled            |
| 20<br>TFEE  | Tx FIFO Empty Interrupt Enable<br>0 Interrupt disabled<br>1 Interrupt enabled                      |
| 21<br>TCFE  | Transmission Cancellation Finished Interrupt Enable<br>0 Interrupt disabled<br>1 Interrupt enabled |
| 22<br>TCE   | Transmission Completed Interrupt Enable<br>0 Interrupt disabled<br>1 Interrupt enabled             |
| 23<br>HPME  | High Priority Message Interrupt Enable<br>0 Interrupt disabled<br>1 Interrupt enabled              |
| 24<br>RF1LE | Rx FIFO 1 Message Lost Interrupt Enable<br>0 Interrupt disabled<br>1 Interrupt enabled             |
| 25<br>RF1FE | Rx FIFO 1 Full Interrupt Enable<br>0 Interrupt disabled<br>1 Interrupt enabled                     |
| 26<br>RF1WE | Rx FIFO 1 Watermark Reached Interrupt Enable<br>0 Interrupt disabled<br>1 Interrupt enabled        |

*Table continues on the next page...*



**M\_CAN\_IE field descriptions (continued)**

| Field       | Description                                                                                 |
|-------------|---------------------------------------------------------------------------------------------|
| 27<br>RF1NE | Rx FIFO 1 New Message Interrupt Enable<br>0 Interrupt disabled<br>1 Interrupt enabled       |
| 28<br>RF0LE | Rx FIFO 0 Message Lost Interrupt Enable<br>0 Interrupt disabled<br>1 Interrupt enabled      |
| 29<br>RF0FE | Rx FIFO 0 Full Interrupt Enable<br>0 Interrupt disabled<br>1 Interrupt enabled              |
| 30<br>RF0WE | Rx FIFO 0 Watermark Reached Interrupt Enable<br>0 Interrupt disabled<br>1 Interrupt enabled |
| 31<br>RF0NE | Rx FIFO 0 New Message Interrupt Enable<br>0 Interrupt disabled<br>1 Interrupt enabled       |

**38.3.17 Interrupt Line Select Register (M\_CAN\_ILS)**

The Interrupt Line Select register assigns an interrupt generated by a specific interrupt flag from the Interrupt Register to one of the two module interrupt lines. For interrupt generation the respective interrupt line has to be enabled via ILE[EINT0] and ILE[EINT1]

0 = Interrupt assigned to interrupt line 0

1 = Interrupt assigned to interrupt line 1

Address: 0h base + 58h offset = 58h

| Bit   | 0 | 1 | 2    | 3    | 4    | 5    | 6   | 7   | 8   | 9    | 10   | 11   | 12   | 13   | 14    | 15   |
|-------|---|---|------|------|------|------|-----|-----|-----|------|------|------|------|------|-------|------|
| R     | 0 |   |      |      |      |      |     |     |     |      |      |      |      |      |       |      |
| W     |   |   | ARAL | PEDL | PEAL | WDIL | BOL | EWL | EPL | ELOL | BEUL | BECL | DRXL | TOOL | MRAFL | TSWL |
| Reset | 0 | 0 | 0    | 0    | 0    | 0    | 0   | 0   | 0   | 0    | 0    | 0    | 0    | 0    | 0     | 0    |

## Memory Map and Register Description

| Bit   | 16   | 17   | 18    | 19    | 20   | 21   | 22  | 23   | 24    | 25    | 26    | 27    | 28    | 29    | 30    | 31    |
|-------|------|------|-------|-------|------|------|-----|------|-------|-------|-------|-------|-------|-------|-------|-------|
| R     | TEFL | TEFL | TEFWL | TEFNL | TFEL | TCFL | TCL | HPML | RF1LL | RF1FL | RF1WL | RF1NL | RF0LL | RF0FL | RF0WL | RF0NL |
| W     |      |      |       |       |      |      |     |      |       |       |       |       |       |       |       |       |
| Reset | 0    | 0    | 0     | 0     | 0    | 0    | 0   | 0    | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

### M\_CAN\_ILS field descriptions

| Field           | Description                                                                                                                                  |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| 0–1<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                      |
| 2<br>ARAL       | Access to Reserved Address Line<br>0 Interrupt assigned to M_CAN interrupt line 0<br>1 Interrupt assigned to M_CAN interrupt line 1          |
| 3<br>PEDL       | Protocol Error in Data Phase Line<br>0 Interrupt assigned to M_CAN interrupt line 0<br>1 Interrupt assigned to M_CAN interrupt line 1        |
| 4<br>PEAL       | Protocol Error in Arbitration Phase Line<br>0 Interrupt assigned to M_CAN interrupt line 0<br>1 Interrupt assigned to M_CAN interrupt line 1 |
| 5<br>WDIL       | Watchdog Interrupt Line<br>0 Interrupt assigned to M_CAN interrupt line 0<br>1 Interrupt assigned to M_CAN interrupt line 1                  |
| 6<br>BOL        | Bus_Off Status Interrupt Line<br>0 Interrupt assigned to M_CAN interrupt line 0<br>1 Interrupt assigned to M_CAN interrupt line 1            |
| 7<br>EWL        | Warning Status Interrupt Line<br>0 Interrupt assigned to M_CAN interrupt line 0<br>1 Interrupt assigned to M_CAN interrupt line 1            |
| 8<br>EPL        | Error Passive Interrupt Line<br>0 Interrupt assigned to M_CAN interrupt line 0<br>1 Interrupt assigned to M_CAN interrupt line 1             |
| 9<br>ELOL       | Error Logging Overflow Interrupt Line<br>0 Interrupt assigned to M_CAN interrupt line 0<br>1 Interrupt assigned to M_CAN interrupt line 1    |
| 10<br>BEUL      | Bit Error Uncorrected Interrupt Line<br>0 Interrupt assigned to M_CAN interrupt line 0<br>1 Interrupt assigned to M_CAN interrupt line 1     |
| 11<br>BECL      | Bit Error Corrected Interrupt Line                                                                                                           |

Table continues on the next page...

**M\_CAN\_ILS field descriptions (continued)**

| <b>Field</b> | <b>Description</b>                                                                                                                                           |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
|              | 0 Interrupt assigned to M_CAN interrupt line 0<br>1 Interrupt assigned to M_CAN interrupt line 1                                                             |
| 12<br>DRXL   | Message stored to Dedicated Rx Buffer Interrupt Line<br><br>0 Interrupt assigned to M_CAN interrupt line 0<br>1 Interrupt assigned to M_CAN interrupt line 1 |
| 13<br>TOOL   | Timeout Occurred Interrupt Line<br><br>0 Interrupt assigned to M_CAN interrupt line 0<br>1 Interrupt assigned to M_CAN interrupt line 1                      |
| 14<br>MRAFL  | Message RAM Access Failure Interrupt Line<br><br>0 Interrupt assigned to M_CAN interrupt line 0<br>1 Interrupt assigned to M_CAN interrupt line 1            |
| 15<br>TSWL   | Timestamp Wraparound Interrupt Line<br><br>0 Interrupt assigned to M_CAN interrupt line 0<br>1 Interrupt assigned to M_CAN interrupt line 1                  |
| 16<br>TEFLL  | Tx Event FIFO Element Lost Interrupt Line<br><br>0 Interrupt assigned to M_CAN interrupt line 0<br>1 Interrupt assigned to M_CAN interrupt line 1            |
| 17<br>TEFFL  | Tx Event FIFO Full Interrupt Line<br><br>0 Interrupt assigned to M_CAN interrupt line 0<br>1 Interrupt assigned to M_CAN interrupt line 1                    |
| 18<br>TEFWL  | Tx Event FIFO Watermark Reached Interrupt Line<br><br>0 Interrupt assigned to M_CAN interrupt line 0<br>1 Interrupt assigned to M_CAN interrupt line 1       |
| 19<br>TEFNL  | Tx Event FIFO New Entry Interrupt Line<br><br>0 Interrupt assigned to M_CAN interrupt line 0<br>1 Interrupt assigned to M_CAN interrupt line 1               |
| 20<br>TFEL   | Tx FIFO Empty Interrupt Line<br><br>0 Interrupt assigned to M_CAN interrupt line 0<br>1 Interrupt assigned to M_CAN interrupt line 1                         |
| 21<br>TCFL   | Transmission Cancellation Finished Interrupt Line<br><br>0 Interrupt assigned to M_CAN interrupt line 0<br>1 Interrupt assigned to M_CAN interrupt line 1    |
| 22<br>TCL    | Transmission Completed Interrupt Line<br><br>0 Interrupt assigned to M_CAN interrupt line 0<br>1 Interrupt assigned to M_CAN interrupt line 1                |
| 23<br>HPML   | High Priority Message Interrupt Line                                                                                                                         |

*Table continues on the next page...*

**M\_CAN\_ILS field descriptions (continued)**

| Field       | Description                                                                                                                                        |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
|             | 0 Interrupt assigned to M_CAN interrupt line 0<br>1 Interrupt assigned to M_CAN interrupt line 1                                                   |
| 24<br>RF1LL | Rx FIFO 1 Message Lost Interrupt Line<br><br>0 Interrupt assigned to M_CAN interrupt line 0<br>1 Interrupt assigned to M_CAN interrupt line 1      |
| 25<br>RF1FL | Rx FIFO 1 Full Interrupt Line<br><br>0 Interrupt assigned to M_CAN interrupt line 0<br>1 Interrupt assigned to M_CAN interrupt line 1              |
| 26<br>RF1WL | Rx FIFO 1 Watermark Reached Interrupt Line<br><br>0 Interrupt assigned to M_CAN interrupt line 0<br>1 Interrupt assigned to M_CAN interrupt line 1 |
| 27<br>RF1NL | Rx FIFO 1 New Message Interrupt Line<br><br>0 Interrupt assigned to M_CAN interrupt line 0<br>1 Interrupt assigned to M_CAN interrupt line 1       |
| 28<br>RF0LL | Rx FIFO 0 Message Lost Interrupt Line<br><br>0 Interrupt assigned to M_CAN interrupt line 0<br>1 Interrupt assigned to M_CAN interrupt line 1      |
| 29<br>RF0FL | Rx FIFO 0 Full Interrupt Line<br><br>0 Interrupt assigned to M_CAN interrupt line 0<br>1 Interrupt assigned to M_CAN interrupt line 1              |
| 30<br>RF0WL | Rx FIFO 0 Watermark Reached Interrupt Line<br><br>0 Interrupt assigned to M_CAN interrupt line 0<br>1 Interrupt assigned to M_CAN interrupt line 1 |
| 31<br>RF0NL | Rx FIFO 0 New Message Interrupt Line<br><br>0 Interrupt assigned to M_CAN interrupt line 0<br>1 Interrupt assigned to M_CAN interrupt line 1       |

### 38.3.18 Interrupt Line Enable Register (M\_CAN\_ILE)

Each of the two interrupt lines to the CPU can be enabled / disabled separately by programming bits EINT0 and EINT1.

Address: 0h base + 5Ch offset = 5Ch

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |       |       |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--|--|--|--|--|--|--|--|--|--|--|--|--|-------|-------|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |  |  |  |  |  |  |  |  |  |  |  |  |  |       |       |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |       |       |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |       |       |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |  |  |  |  |  |  |  |  |  |  |  |  |  |       |       |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |  |  |  |  |  |  |  |  |  |  |  |  |  |       |       |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  | EINT1 | EINT0 |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |       |       |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |  |  |  |  |  |  |  |  |  |  |  |  |  |       |       |

#### M\_CAN\_ILE field descriptions

| Field            | Description                                                                                          |
|------------------|------------------------------------------------------------------------------------------------------|
| 0–29<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.              |
| 30<br>EINT1      | Enable Interrupt Line 1<br><br>0 M_CAN interrupt line 1 disabled<br>1 M_CAN interrupt line 1 enabled |
| 31<br>EINT0      | Enable Interrupt Line 0<br><br>0 M_CAN interrupt line 0 disabled<br>1 M_CAN interrupt line 0 enabled |

### 38.3.19 Global Filter Configuration Register (M\_CAN\_GFC)

Global settings for Message ID filtering. The Global Filter Configuration controls the filter path for standard and extended messages as described in the [Standard Message ID Filtering](#) and [Extended Message ID Filtering](#).

Address: 0h base + 80h offset = 80h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |    |    |    |    |    |    |    |      |    |      |    |      |    |      |    |
|-------|----|----|----|----|----|----|----|----|------|----|------|----|------|----|------|----|
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24   | 25 | 26   | 27 | 28   | 29 | 30   | 31 |
| R     | 0  |    |    |    |    |    |    |    | ANFS |    | ANFE |    | RRFS |    | RRFE |    |
| W     |    |    |    |    |    |    |    |    |      |    |      |    |      |    |      |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0  | 0    | 0  | 0    | 0  | 0    | 0  |

#### M\_CAN\_GFC field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                 |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–25<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                     |
| 26–27<br>ANFS    | Accept Non-matching Frames Standard<br>Defines how received messages with 11-bit IDs that do not match any element of the filter list are treated.<br><b>NOTE:</b> This field has Protected Write status.<br><br>00 Accept in Rx FIFO 0<br>01 Accept in Rx FIFO 1<br>10 Reject<br>11 Reject |
| 28–29<br>ANFE    | Accept Non-matching Frames Extended<br>Defines how received messages with 29-bit IDs that do not match any element of the filter list are treated.<br><b>NOTE:</b> This field has Protected Write status.<br><br>00 Accept in Rx FIFO 0<br>01 Accept in Rx FIFO 1<br>10 Reject<br>11 Reject |
| 30<br>RRFS       | Reject Remote Frames Standard<br><br><b>NOTE:</b> This field has Protected Write status.<br><br>0 Filter remote frames with 11-bit standard IDs<br>1 Reject all remote frames with 11-bit standard IDs                                                                                      |

Table continues on the next page...

**M\_CAN\_GFC field descriptions (continued)**

| Field      | Description                                                                                                                                                                                            |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31<br>RRFE | Reject Remote Frames Extended<br><br><b>NOTE:</b> This field has Protected Write status.<br><br>0 Filter remote frames with 29-bit extended IDs<br>1 Reject all remote frames with 29-bit extended IDs |

**38.3.20 Standard ID Filter Configuration Register (M\_CAN\_SIDFC)**

Settings for 11-bit standard Message ID filtering. The Standard ID Filter Configuration controls the filter path for the standard messages as described in [Standard Message ID Filtering](#).

Address: 0h base + 84h offset = 84h

|       |   |   |   |   |   |   |   |   |     |   |    |    |    |    |    |    |       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|-----|---|----|----|----|----|----|----|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8   | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16    | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   | LSS |   |    |    |    |    |    |    | FLSSA |    |    |    |    |    |    |    |    |    |    |    | 0  |    |    |    |
| W     |   |   |   |   |   |   |   |   |     |   |    |    |    |    |    |    |       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0     | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**M\_CAN\_SIDFC field descriptions**

| Field             | Description                                                                                                                                                                                                                        |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–7<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                            |
| 8–15<br>LSS       | List Size Standard<br><br><b>NOTE:</b> This field has Protected Write status.<br><br>0 No standard Message ID filter<br>1–128 Number of standard Message ID filter elements<br>>128 Values greater than 128 are interpreted as 128 |
| 16–29<br>FLSSA    | Filter List Standard Start Address<br><br>Start address of standard Message ID filter list (32-bit word address, see <a href="#">Message RAM</a> ).<br><br><b>NOTE:</b> This field has Protected Write status.                     |
| 30–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                            |

### 38.3.21 Extended ID Filter Configuration Register (M\_CAN\_XIDFC)

Settings for 29-bit extended Message ID filtering. The Extended ID Filter Configuration controls the filter path for the standard messages as described in [Extended Message ID Filtering](#).

Address: 0h base + 88h offset = 88h

|       |   |   |   |   |   |   |   |   |   |     |    |    |    |    |    |    |       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|-----|----|----|----|----|----|----|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9   | 10 | 11 | 12 | 13 | 14 | 15 | 16    | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   | LSE |    |    |    |    |    |    | FLESA |    |    |    |    |    |    |    |    |    |    |    |    |    | 0  |    |
| W     |   |   |   |   |   |   |   |   |   |     |    |    |    |    |    |    |       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0     | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

#### M\_CAN\_XIDFC field descriptions

| Field             | Description                                                                                                                                                                                                                             |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–8<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                 |
| 9–15<br>LSE       | List Size Extended<br><br><b>NOTE:</b> This field has Protected Write status.<br><br>0      No extended Message ID filter<br>1–64   Number of extended Message ID filter elements<br>>64   Values greater than 64 are interpreted as 64 |
| 16–29<br>FLESA    | Filter List Extended Start Address<br><br>Start address of extended Message ID filter list (32-bit word address, see <a href="#">Message RAM</a> ).<br><br><b>NOTE:</b> This field has Protected Write status.                          |
| 30–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                 |

### 38.3.22 Extended ID and Mask Register (M\_CAN\_XIDAM)

Address: 0h base + 90h offset = 90h

|       |   |   |   |      |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|------|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3    | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   | EIDM |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |      |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 1    | 1 | 1 | 1 | 1 | 1 | 1 | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  |    |



## M\_CAN\_XIDAM field descriptions

| Field           | Description                                                                                                                                                                                                                                                                                                                    |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–2<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                        |
| 3–31<br>EIDM    | Extended ID Mask<br><br>For acceptance filtering of extended frames the Extended ID and Mask is ANDed with the Message ID of a received frame. Intended for masking of 29-bit IDs in SAE J1939. With the reset value of all bits set to one the mask is not active.<br><br><b>NOTE:</b> This field has Protected Write status. |

## 38.3.23 High Priority Message Status Register (M\_CAN\_HPMS)

This register is updated every time a Message ID filter element configured to generate a priority event matches. This can be used to monitor the status of incoming high priority messages and to enable fast access to these messages.

Address: 0h base + 94h offset = 94h

|       |      |      |    |    |    |    |    |    |     |    |      |    |    |    |    |    |
|-------|------|------|----|----|----|----|----|----|-----|----|------|----|----|----|----|----|
| Bit   | 0    | 1    | 2  | 3  | 4  | 5  | 6  | 7  | 8   | 9  | 10   | 11 | 12 | 13 | 14 | 15 |
| R     | 0    |      |    |    |    |    |    |    |     |    |      |    |    |    |    |    |
| W     |      |      |    |    |    |    |    |    |     |    |      |    |    |    |    |    |
| Reset | 0    | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0  | 0    | 0  | 0  | 0  | 0  | 0  |
| Bit   | 16   | 17   | 18 | 19 | 20 | 21 | 22 | 23 | 24  | 25 | 26   | 27 | 28 | 29 | 30 | 31 |
| R     | FLST | FIDX |    |    |    |    |    |    | MSI |    | BIDX |    |    |    |    |    |
| W     |      |      |    |    |    |    |    |    |     |    |      |    |    |    |    |    |
| Reset | 0    | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0  | 0    | 0  | 0  | 0  | 0  | 0  |

## M\_CAN\_HPMS field descriptions

| Field            | Description                                                                                                                          |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                              |
| 16<br>FLST       | Filter List<br><br>Indicates the filter list of the matching filter element.<br><br>0 Standard Filter List<br>1 Extended Filter List |
| 17–23<br>FIDX    | Filter Index<br><br>Index of matching filter element. Range is 0 to SIDFC[LSS] - 1 resp. XIDFC[LSE] - 1.                             |
| 24–25<br>MSI     | Message Storage Indicator                                                                                                            |

Table continues on the next page...

**M\_CAN\_HPMS field descriptions (continued)**

| Field         | Description                                                                                               |
|---------------|-----------------------------------------------------------------------------------------------------------|
|               | 00 No FIFO selected<br>01 FIFO message lost<br>10 Message stored in FIFO 0<br>11 Message stored in FIFO 1 |
| 26–31<br>BIDX | Buffer Index<br>Index of Rx FIFO element to which the message was stored. Only valid when MSI[1] = 1.     |

**38.3.24 New Data 1 Register (M\_CAN\_NDAT1)**

Address: 0h base + 98h offset = 98h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**M\_CAN\_NDAT1 field descriptions**

| Field       | Description                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>ND1 | New Data[0:31]<br><br>The register holds the New Data flags of Rx Buffers 0 to 31. The flags are set when the respective Rx Buffer has been updated from a received frame. The flags remain set until the CPU clears them. A flag is cleared by writing a 1 to the corresponding bit position. Writing a 0 has no effect. A hard reset will clear the register.<br><br>0 Rx Buffer not updated<br>1 Rx Buffer updated from new message |

**38.3.25 New Data 2 Register (M\_CAN\_NDAT2)**

Address: 0h base + 9Ch offset = 9Ch

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**M\_CAN\_NDAT2 field descriptions**

| Field       | Description                                                                                                                                                                                                                                                                                                                                                       |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>ND2 | New Data[32:63]<br><br>The register holds the New Data flags of Rx Buffers 32 to 63. The flags are set when the respective Rx Buffer has been updated from a received frame. The flags remain set until the CPU clears them. A flag is cleared by writing a 1 to the corresponding bit position. Writing a 0 has no effect. A hard reset will clear the register. |

**M\_CAN\_NDAT2 field descriptions (continued)**

| Field | Description                        |
|-------|------------------------------------|
| 0     | Rx Buffer not updated              |
| 1     | Rx Buffer updated from new message |

**38.3.26 Rx FIFO 0 Configuration Register (M\_CAN\_RXF0C)**

Address: 0h base + A0h offset = A0h

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
| R     |    |    |    |    |    |    |    |    | 0  |    |    |    |    |    |    |    |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0  |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**M\_CAN\_RXF0C field descriptions**

| Field         | Description                                                                                                                                                                                                                   |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>F0OM     | FIFO 0 Operation Mode<br>FIFO 0 can be operated in blocking or in overwrite mode (see <a href="#">Rx FIFOs</a> ).<br><b>NOTE:</b> This field has Protected Write status.<br>0 FIFO 0 blocking mode<br>1 FIFO 0 overwrite mode |
| 1–7<br>F0WM   | Rx FIFO 0 Watermark<br><b>NOTE:</b> This field has Protected Write status.<br>0 Watermark interrupt disabled<br>1–64 Level for Rx FIFO 0 watermark interrupt (IR[RF0W])<br>>64 Watermark interrupt disabled                   |
| 8<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                       |
| 9–15<br>F0S   | Rx FIFO 0 Size<br>The Rx FIFO 0 elements are indexed from 0 to F0S-1.<br><b>NOTE:</b> This field has Protected Write status.                                                                                                  |

*Table continues on the next page...*

**M\_CAN\_RXF0C field descriptions (continued)**

| Field             | Description                                                                                                                                                                                  |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0                 | No Rx FIFO 0                                                                                                                                                                                 |
| 1-64              | Number of Rx FIFO 0 elements                                                                                                                                                                 |
| >64               | Values greater than 64 are interpreted as 64                                                                                                                                                 |
| 16–29<br>F0SA     | Rx FIFO 0 Start Address<br><br>Start address of Rx FIFO 0 in Message RAM (32-bit word address, see <a href="#">Message RAM</a> ).<br><br><b>NOTE:</b> This field has Protected Write status. |
| 30–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                      |

**38.3.27 Rx FIFO 0 Status Register (M\_CAN\_RXF0S)**

Address: 0h base + A4h offset = A4h

|       |   |   |   |   |   |   |      |     |   |   |      |    |    |    |    |    |
|-------|---|---|---|---|---|---|------|-----|---|---|------|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6    | 7   | 8 | 9 | 10   | 11 | 12 | 13 | 14 | 15 |
| R     | 0 |   |   |   |   |   | RF0L | F0F | 0 |   | F0PI |    |    |    |    |    |
| W     |   |   |   |   |   |   |      |     |   |   |      |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0    | 0   | 0 | 0 | 0    | 0  | 0  | 0  | 0  | 0  |

|       |    |    |      |    |    |    |    |    |    |    |      |    |    |    |    |    |
|-------|----|----|------|----|----|----|----|----|----|----|------|----|----|----|----|----|
| Bit   | 16 | 17 | 18   | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26   | 27 | 28 | 29 | 30 | 31 |
| R     | 0  |    | F0GI |    |    |    |    |    | 0  |    | F0FL |    |    |    |    |    |
| W     |    |    |      |    |    |    |    |    |    |    |      |    |    |    |    |    |
| Reset | 0  | 0  | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0  | 0  | 0  | 0  | 0  |

**M\_CAN\_RXF0S field descriptions**

| Field           | Description                                                                                                                                                                                                                                                                                                                                          |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–5<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                              |
| 6<br>RF0L       | Rx FIFO 0 Message Lost<br><br>This bit is a copy of interrupt flag IR[RF0L]. When IR[RF0L] is reset, this bit is also reset.<br><br><b>NOTE:</b> Overwriting the oldest message when RXF0C.FOOM = '1' will not set this flag.<br><br>0 No Rx FIFO 0 message lost<br>1 Rx FIFO 0 message lost, also set after write attempt to Rx FIFO 0 of size zero |
| 7<br>F0F        | Rx FIFO 0 Full<br><br>0 Rx FIFO 0 not full<br>1 Rx FIFO 0 full                                                                                                                                                                                                                                                                                       |
| 8–9<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                              |
| 10–15<br>F0PI   | Rx FIFO 0 Put Index<br><br>Rx FIFO 0 write index pointer, range 0 to 63                                                                                                                                                                                                                                                                              |

*Table continues on the next page...*

**M\_CAN\_RXF0S field descriptions (continued)**

| Field             | Description                                                                             |
|-------------------|-----------------------------------------------------------------------------------------|
| 16–17<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 18–23<br>F0GI     | Rx FIFO 0 Get Index<br>Rx FIFO 0 read index pointer, range 0 to 63.                     |
| 24<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 25–31<br>F0FL     | Rx FIFO 0 Fill Level<br>Number of elements stored in Rx FIFO 0, range 0 to 64.          |

**38.3.28 Rx FIFO 0 Acknowledge Register (M\_CAN\_RXF0A)**

Address: 0h base + A8h offset = A8h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25   | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | FOAI |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0  | 0  | 0  | 0  | 0  |    |

**M\_CAN\_RXF0A field descriptions**

| Field            | Description                                                                                                                                                                                                                                                                                                |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–25<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                    |
| 26–31<br>F0AI    | Rx FIFO 0 Acknowledge Index<br><br>After the CPU has read a message or a sequence of messages from Rx FIFO 0 it has to write the buffer index of the last element read from Rx FIFO 0 to F0AI. This will set the Rx FIFO 0 Get Index RXF0S[F0GI] to F0AI + 1 and update the FIFO 0 Fill Level RXF0S[F0FL]. |

**38.3.29 Rx Buffer Configuration Register (M\_CAN\_RXBC)**

Address: 0h base + ACh offset = ACh

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16   | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | RBSA |    |    |    |    |    |    |    |    |    |    |    |    | 0  |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | RBSA |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**M\_CAN\_RXBC field descriptions**

| Field            | Description                                                                             |
|------------------|-----------------------------------------------------------------------------------------|
| 0–15<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |

*Table continues on the next page...*

**M\_CAN\_RXBC field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                               |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 16–29<br>RBSA     | Rx Buffer Start Address<br><br>Configures the start address of the Rx Buffers section in the Message RAM (32-bit word address). Also used to reference debug messages A, B, C.<br><br><b>NOTE:</b> This field has Protected Write status. |
| 30–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                   |

**38.3.30 Rx FIFO 1 Configuration Register (M\_CAN\_RXF1C)**

Address: 0h base + B0h offset = B0h

|       |      |      |    |    |    |    |    |    |    |     |    |    |    |    |    |    |
|-------|------|------|----|----|----|----|----|----|----|-----|----|----|----|----|----|----|
| Bit   | 0    | 1    | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9   | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | F1OM | F1WM |    |    |    |    |    |    | 0  | F1S |    |    |    |    |    |    |
| W     |      |      |    |    |    |    |    |    |    |     |    |    |    |    |    |    |
| Reset | 0    | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0  | 0  | 0  | 0  | 0  | 0  |
| Bit   | 16   | 17   | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25  | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | F1SA |      |    |    |    |    |    |    |    |     |    |    |    |    | 0  |    |
| W     |      |      |    |    |    |    |    |    |    |     |    |    |    |    |    |    |
| Reset | 0    | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0  | 0  | 0  | 0  | 0  | 0  |

**M\_CAN\_RXF1C field descriptions**

| Field         | Description                                                                                                                                                                                                                               |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>F1OM     | FIFO 1 Operation Mode<br><br>FIFO 1 can be operated in blocking or in overwrite mode (see <a href="#">Rx FIFOs</a> ).<br><br><b>NOTE:</b> This field has Protected Write status.<br><br>0 FIFO 1 blocking mode<br>1 FIFO 1 overwrite mode |
| 1–7<br>F1WM   | Rx FIFO 1 Watermark<br><br><b>NOTE:</b> This field has Protected Write status.<br><br>0 Watermark interrupt disabled<br>1–64 Level for Rx FIFO 1 watermark interrupt (IR[RF1W])<br>>64 Watermark interrupt disabled                       |
| 8<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                   |

*Table continues on the next page...*

**M\_CAN\_RXF1C field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                       |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 9–15<br>F1S       | <p>Rx FIFO 1 Size</p> <p>The Rx FIFO 1 elements are indexed from 0 to F1S - 1.</p> <p><b>NOTE:</b> This field has Protected Write status.</p> <p>0 No Rx FIFO 1<br/>1-64 Number of Rx FIFO 1 elements<br/>&gt;64 Values greater than 64 are interpreted as 64</p> |
| 16–29<br>F1SA     | <p>Rx FIFO 1 Start Address</p> <p>Start address of Rx FIFO 1 in Message RAM (32-bit word address, see <a href="#">Message RAM</a>).</p> <p><b>NOTE:</b> This field has Protected Write status.</p>                                                                |
| 30–31<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                |

**38.3.31 Rx FIFO 1 Status Register (M\_CAN\_RXF1S)**

Address: 0h base + B4h offset = B4h

|       |          |    |    |    |    |    |      |      |    |    |    |    |    |    |      |    |
|-------|----------|----|----|----|----|----|------|------|----|----|----|----|----|----|------|----|
| Bit   | 0        | 1  | 2  | 3  | 4  | 5  | 6    | 7    | 8  | 9  | 10 | 11 | 12 | 13 | 14   | 15 |
| R     | DMS      |    |    | 0  |    |    | RF1L | F1F  |    | 0  |    |    |    |    | F1PI |    |
| W     |          |    |    |    |    |    |      |      |    |    |    |    |    |    |      |    |
| Reset | 0        | 0  | 0  | 0  | 0  | 0  | 0    | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0  |
| Bit   | 16       | 17 | 18 | 19 | 20 | 21 | 22   | 23   | 24 | 25 | 26 | 27 | 28 | 29 | 30   | 31 |
| R     | Reserved |    |    |    |    |    |      | F1GI |    | 0  |    |    |    |    | F1FL |    |
| W     |          |    |    |    |    |    |      |      |    |    |    |    |    |    |      |    |
| Reset | 0        | 0  | 0  | 0  | 0  | 0  | 0    | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0  |

**M\_CAN\_RXF1S field descriptions**

| Field           | Description                                                                                                                                                                                                                                   |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–1<br>DMS      | <p>Debug Message Status</p> <p>00 Idle state, wait for reception of debug messages, DMA request is cleared<br/>01 Debug message A received<br/>10 Debug messages A, B received<br/>11 Debug messages A, B, C received, DMA request is set</p> |
| 2–5<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                            |
| 6<br>RF1L       | <p>Rx FIFO 1 Message Lost</p> <p>This bit is a copy of interrupt flag IR[RF1L]. When IR[RF1L] is reset, this bit is also reset.</p> <p><b>NOTE:</b> Overwriting the oldest message when RXF1C[F1OM] = 1 will not set this flag.</p>           |

*Table continues on the next page...*

**M\_CAN\_RXF1S field descriptions (continued)**

| Field             | Description                                                                                                     |
|-------------------|-----------------------------------------------------------------------------------------------------------------|
|                   | 0 No Rx FIFO 1 message lost<br>1 Rx FIFO 1 message lost, also set after write attempt to Rx FIFO 1 of size zero |
| 7<br>F1F          | Rx FIFO 1 Full<br><br>0 Rx FIFO 1 not full<br>1 Rx FIFO 1 full                                                  |
| 8–9<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                         |
| 10–15<br>F1PI     | Rx FIFO 1 Put Index<br><br>Rx FIFO 1 write index pointer, range 0 to 63.                                        |
| 16–17<br>Reserved | This field is reserved.                                                                                         |
| 18–23<br>F1GI     | Rx FIFO 1 Get Index<br><br>Rx FIFO 1 read index pointer, range 0 to 63.                                         |
| 24<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                         |
| 25–31<br>F1FL     | Rx FIFO 1 Fill Level<br><br>Number of elements stored in Rx FIFO 1, range 0 to 64.                              |

**38.3.32 Rx FIFO 1 Acknowledge Register (M\_CAN\_RXF1A)**

Address: 0h base + B8h offset = B8h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25   | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | F1AI |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0  | 0  | 0  | 0  | 0  | 0  |

**M\_CAN\_RXF1A field descriptions**

| Field            | Description                                                                                                                                                                                                                                                                                                |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–25<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                    |
| 26–31<br>F1AI    | Rx FIFO 1 Acknowledge Index<br><br>After the CPU has read a message or a sequence of messages from Rx FIFO 1 it has to write the buffer index of the last element read from Rx FIFO 1 to F1AI. This will set the Rx FIFO 1 Get Index RXF1S[F1GI] to F1AI + 1 and update the FIFO 1 Fill Level RXF1S[F1FL]. |



### 38.3.33 Rx Buffer / FIFO Element Size Configuration Register (M\_CAN\_RXESC)

Configures the number of data bytes belonging to an Rx Buffer / Rx FIFO element. Data field sizes >8 bytes are intended for CAN FD operation only.

#### NOTE

In case the data field size of an accepted CAN frame exceeds the data field size configured for the matching Rx Buffer or Rx FIFO, only the number of bytes as configured by RXESC are stored to the Rx Buffer resp. Rx FIFO element. The rest of the frame's data field is ignored.

Address: 0h base + BCh offset = BCh

|       |    |    |    |    |      |    |    |    |    |      |    |    |    |    |      |    |
|-------|----|----|----|----|------|----|----|----|----|------|----|----|----|----|------|----|
| Bit   | 0  | 1  | 2  | 3  | 4    | 5  | 6  | 7  | 8  | 9    | 10 | 11 | 12 | 13 | 14   | 15 |
| R     | 0  |    |    |    |      |    |    |    |    |      |    |    |    |    |      |    |
| W     |    |    |    |    |      |    |    |    |    |      |    |    |    |    |      |    |
| Reset | 0  | 0  | 0  | 0  | 0    | 0  | 0  | 0  | 0  | 0    | 0  | 0  | 0  | 0  | 0    | 0  |
| Bit   | 16 | 17 | 18 | 19 | 20   | 21 | 22 | 23 | 24 | 25   | 26 | 27 | 28 | 29 | 30   | 31 |
| R     | 0  |    |    |    | RBDS |    |    |    | 0  | F1DS |    |    |    | 0  | F0DS |    |
| W     |    |    |    |    |      |    |    |    |    |      |    |    |    |    |      |    |
| Reset | 0  | 0  | 0  | 0  | 0    | 0  | 0  | 0  | 0  | 0    | 0  | 0  | 0  | 0  | 0    | 0  |

#### M\_CAN\_RXESC field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                             |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–20<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                 |
| 21–23<br>RBDS    | Rx Buffer Data Field Size<br><br><b>NOTE:</b> This field has Protected Write status.<br><br>000 8 byte data field<br>001 12 byte data field<br>010 16 byte data field<br>011 20 byte data field<br>100 24 byte data field<br>101 32 byte data field<br>110 48 byte data field<br>111 64 byte data field |
| 24<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                 |
| 25–27<br>F1DS    | Rx FIFO 1 Data Field Size<br><br><b>NOTE:</b> This field has Protected Write status.                                                                                                                                                                                                                    |

Table continues on the next page...

**M\_CAN\_RXESC field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                                                             |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | 000 8 byte data field<br>001 12 byte data field<br>010 16 byte data field<br>011 20 byte data field<br>100 24 byte data field<br>101 32 byte data field<br>110 48 byte data field<br>111 64 byte data field                                                                                             |
| 28<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                 |
| 29–31<br>F0DS  | Rx FIFO 0 Data Field Size<br><br><b>NOTE:</b> This field has Protected Write status.<br><br>000 8 byte data field<br>001 12 byte data field<br>010 16 byte data field<br>011 20 byte data field<br>100 24 byte data field<br>101 32 byte data field<br>110 48 byte data field<br>111 64 byte data field |

### 38.3.34 Tx Buffer Configuration Register (M\_CAN\_TXBC)

#### NOTE

Be aware that the sum of TFQS and NDTB may be not greater than 32. There is no check for erroneous configurations. The Tx Buffers section in the Message RAM starts with the dedicated Tx Buffers.

Address: 0h base + C0h offset = C0h

|       |      |      |      |    |    |    |    |    |    |      |    |    |    |    |    |    |
|-------|------|------|------|----|----|----|----|----|----|------|----|----|----|----|----|----|
| Bit   | 0    | 1    | 2    | 3  | 4  | 5  | 6  | 7  | 8  | 9    | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0    | TFQM | TFQS |    |    |    |    |    | 0  | NDTB |    |    |    |    |    |    |
| W     |      |      |      |    |    |    |    |    |    |      |    |    |    |    |    |    |
| Reset | 0    | 0    | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0  | 0  | 0  | 0  | 0  | 0  |
| Bit   | 16   | 17   | 18   | 19 | 20 | 21 | 22 | 23 | 24 | 25   | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | TBSA |      |      |    |    |    |    |    |    |      |    |    |    |    | 0  |    |
| W     |      |      |      |    |    |    |    |    |    |      |    |    |    |    |    |    |
| Reset | 0    | 0    | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0  | 0  | 0  | 0  | 0  | 0  |

#### M\_CAN\_TXBC field descriptions

| Field           | Description                                                                                                                                                                                                           |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                               |
| 1<br>TFQM       | Tx FIFO/Queue Mode<br><br><b>NOTE:</b> This field has Protected Write status.<br><br>0 Tx FIFO operation<br>1 Tx Queue operation                                                                                      |
| 2–7<br>TFQS     | Transmit FIFO/Queue Size<br><br><b>NOTE:</b> This field has Protected Write status.<br><br>0 No Tx FIFO/Queue<br>1–32 Number of Tx Buffers used for Tx FIFO/Queue<br>>32 Values greater than 32 are interpreted as 32 |
| 8–9<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                               |

Table continues on the next page...

**M\_CAN\_TXBC field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                          |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 10–15<br>NDTB     | Number of Dedicated Transmit Buffers<br><br><b>NOTE:</b> This field has Protected Write status.<br><br>0      No Dedicated Tx Buffers<br>1–32   Number of Dedicated Tx Buffers<br>>32   Values greater than 32 are interpreted as 32 |
| 16–29<br>TBSA     | Tx Buffers Start Address<br><br>Start address of Tx Buffers section in Message RAM (32-bit word address, see <a href="#">Message RAM</a> ).<br><br><b>NOTE:</b> This field has Protected Write status.                               |
| 30–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                              |

**38.3.35 Tx FIFO/Queue Status Register (M\_CAN\_TXFQS)**

The Tx FIFO/Queue status is related to the pending Tx requests listed in register TXBRP. Therefore the effect of Add/Cancellation requests may be delayed due to a running Tx scan (TXBRP not yet updated).

**NOTE**

In case of mixed configurations where dedicated Tx Buffers are combined with a Tx FIFO or a Tx Queue, the Put and Get Indices indicate the number of the Tx Buffer starting with the first dedicated Tx Buffers.

Example: For a configuration of 12 dedicated Tx Buffers and a Tx FIFO of 20 Buffers a Put Index of 15 points to the fourth buffer of the Tx FIFO.

Address: 0h base + C4h offset = C4h

|       |   |   |   |   |   |   |   |   |   |      |    |       |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|------|----|-------|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9    | 10 | 11    | 12 | 13 | 14 | 15 |
| R     | 0 |   |   |   |   |   |   |   |   | TFQF |    | TFQPI |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |      |    |       |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0    | 0  | 0     | 0  | 0  | 0  | 0  |

|       |    |    |    |      |    |    |    |    |    |    |    |      |    |    |    |    |  |
|-------|----|----|----|------|----|----|----|----|----|----|----|------|----|----|----|----|--|
| Bit   | 16 | 17 | 18 | 19   | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27   | 28 | 29 | 30 | 31 |  |
| R     | 0  |    |    | TFGI |    |    |    |    |    | 0  |    | TFFL |    |    |    |    |  |
| W     |    |    |    |      |    |    |    |    |    |    |    |      |    |    |    |    |  |
| Reset | 0  | 0  | 0  | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0  | 0  | 0  | 0  |  |

**M\_CAN\_TXFQS field descriptions**

| Field             | Description                                                                                                                                                                   |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–9<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                       |
| 10<br>TFQF        | Tx FIFO/Queue Full<br><br>0 Tx FIFO/Queue not full<br>1 Tx FIFO/Queue full                                                                                                    |
| 11–15<br>TFQPI    | Tx FIFO/Queue Put Index<br><br>Tx FIFO/Queue write index pointer, range 0 to 31.                                                                                              |
| 16–18<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                       |
| 19–23<br>TFGI     | Tx FIFO Get Index<br><br>Tx FIFO read index pointer, range 0 to 31. Read as zero when Tx Queue operation is configured (TXBC[TFQM] = 1).                                      |
| 24–25<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                       |
| 26–31<br>TFFL     | Tx FIFO Free Level<br><br>Number of consecutive free Tx FIFO elements starting from TFGI, range 0 to 32. Read as zero when Tx Queue operation is configured (TXBC[TFQM] = 1). |

**38.3.36 Tx Buffer Element Size Configuration (M\_CAN\_TXESC)**

Configures the number of data bytes belonging to a Tx Buffer element. Data field sizes > 8 bytes are intended for CAN FD operation only.

Address: 0h base + C8h offset = C8h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28   | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | TBDS |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0  | 0  |    |

**M\_CAN\_TXESC field descriptions**

| Field            | Description                                                                                                                                                                                                                                                                                                                  |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–28<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                      |
| 29–31<br>TBDS    | Tx Buffer Data Field Size<br><br><b>NOTE:</b> In case the data length code DLC of a Tx Buffer element is configured to a value higher than the Tx Buffer data field size TXESC[TBDS], the bytes not defined by the Tx Buffer are transmitted as 0xCC (padding bytes).<br><b>NOTE:</b> This field has Protected Write status. |

*Table continues on the next page...*

**M\_CAN\_TXESC field descriptions (continued)**

| Field | Description        |
|-------|--------------------|
| 000   | 8 byte data field  |
| 001   | 12 byte data field |
| 010   | 16 byte data field |
| 011   | 20 byte data field |
| 100   | 24 byte data field |
| 101   | 32 byte data field |
| 110   | 48 byte data field |
| 111   | 64 byte data field |

**38.3.37 Tx Buffer Request Pending Register (M\_CAN\_TXBRP)****NOTE**

TXBRP bits which are set while a Tx scan is in progress are not considered during this particular Tx scan. In case a cancellation is requested for such a Tx Buffer, this Add Request is cancelled immediately, the corresponding TXBRP bit is reset.

Address: 0h base + CCh offset = CCh

|       |     |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-----|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | TRP |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |     |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**M\_CAN\_TXBRP field descriptions**

| Field       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>TRP | <p>Transmission Request Pending</p> <p>Each Tx Buffer has its own Transmission Request Pending bit. The bits are set via register TXBAR. The bits are reset after a requested transmission has completed or has been cancelled via register TXBCR.</p> <p>TXBRP bits are set only for those Tx Buffers configured via TXBC. After a TXBRP bit has been set, a Tx scan (see <a href="#">Tx Handling</a>) is started to check for the pending Tx request with the highest priority (Tx Buffer with lowest Message ID).</p> <p>A cancellation request resets the corresponding transmission request pending bit of register TXBRP. In case a transmission has already been started when a cancellation is requested, this is done at the end of the transmission, regardless whether the transmission was successful or not. The cancellation request bits are reset directly after the corresponding TXBRP bit has been reset.</p> <p>After a cancellation has been requested, a finished cancellation is signaled via TXBCF</p> <ul style="list-style-type: none"> <li>• after successful transmission together with the corresponding TXBTO bit</li> <li>• when the transmission has not yet been started at the point of cancellation</li> <li>• when the transmission has been aborted due to lost arbitration</li> <li>• when an error occurred during frame transmission</li> </ul> |

**M\_CAN\_TXBRP field descriptions (continued)**

| Field | Description                                                                                                                                                  |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | In DAR mode all transmissions are automatically cancelled if they are not successful. The corresponding TXBCF bit is set for all unsuccessful transmissions. |
| 0     | No transmission request pending                                                                                                                              |
| 1     | Transmission request pending                                                                                                                                 |

**38.3.38 Tx Buffer Add Request Register (M\_CAN\_TXBAR)****NOTE**

If an add request is applied for a Tx Buffer with pending transmission request (corresponding TXBRP bit already set), this add request is ignored.

Address: 0h base + D0h offset = D0h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**M\_CAN\_TXBAR field descriptions**

| Field      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>AR | <p>Add Request</p> <p>Each Tx Buffer has its own Add Request bit. Writing 1 will set the corresponding Add Request bit; writing 0 has no impact. This enables the CPU to set transmission requests for multiple Tx Buffers with one write to TXBAR. TXBAR bits are set only for those Tx Buffers configured via TXBC. When no Tx scan is running, the bits are reset immediately, else the bits remain set until the Tx scan process has completed.</p> <p>0 No transmission request added</p> <p>1 Transmission requested added</p> |

**38.3.39 Tx Buffer Cancellation Request Register (M\_CAN\_TXBCR)**

Address: 0h base + D4h offset = D4h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**M\_CAN\_TXBCR field descriptions**

| Field      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>CR | <p>Cancellation Request</p> <p>Each Tx Buffer has its own Cancellation Request bit. Writing 1 will set the corresponding Cancellation Request bit; writing 0 has no impact. This enables the CPU to set cancellation requests for multiple Tx Buffers with one write to TXBCR. TXBCR bits are set only for those Tx Buffers configured via TXBC. The bits remain set until the corresponding bit of TXBRP is reset.</p> <p>0 No cancellation pending<br/>1 Cancellation pending</p> |

**38.3.40 Tx Buffer Transmission Occurred Register (M\_CAN\_TXBTO)**

Address: 0h base + D8h offset = D8h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    | TO |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**M\_CAN\_TXBTO field descriptions**

| Field      | Description                                                                                                                                                                                                                                                                                                                                                            |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>TO | <p>Transmission Occurred</p> <p>Each Tx Buffer has its own Transmission Occurred bit. The bits are set when the corresponding TXBRP bit is cleared after a successful transmission. The bits are reset when a new transmission is requested by writing 1 to the corresponding bit of register TXBAR.</p> <p>0 No transmission occurred<br/>1 Transmission occurred</p> |

**38.3.41 Tx Buffer Cancellation Finished Register (M\_CAN\_TXBCF)**

Address: 0h base + DCh offset = DCh

|       |    |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | CF |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |    |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**M\_CAN\_TXBCF field descriptions**

| Field      | Description           |
|------------|-----------------------|
| 0–31<br>CF | Cancellation Finished |



**M\_CAN\_TXBCF field descriptions (continued)**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                            |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | Each Tx Buffer has its own Cancellation Finished bit. The bits are set when the corresponding TXBRP bit is cleared after a cancellation was requested via TXBCR. In case the corresponding TXBRP bit was not set at the point of cancellation, CF is set immediately. The bits are reset when a new transmission is requested by writing 1 to the corresponding bit of register TXBAR. |
| 0     | No transmit buffer cancellation                                                                                                                                                                                                                                                                                                                                                        |
| 1     | Transmit buffer cancellation finished                                                                                                                                                                                                                                                                                                                                                  |

**38.3.42 Tx Buffer Transmission Interrupt Enable Register (M\_CAN\_TXBTIE)**

Address: 0h base + E0h offset = E0h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**M\_CAN\_TXBTIE field descriptions**

| Field       | Description                                                                                                                                                            |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>TIE | Transmission Interrupt Enable<br>Each Tx Buffer has its own Transmission Interrupt Enable bit.<br>0 Transmission interrupt disabled<br>1 Transmission interrupt enable |

**38.3.43 Tx Buffer Cancellation Finished Interrupt Enable Register (M\_CAN\_TXBCIE)**

Address: 0h base + E4h offset = E4h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**M\_CAN\_TXBCIE field descriptions**

| Field        | Description                                                                                                                                                                                                 |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>CFIE | Cancellation Finished Interrupt Enable<br>Each Tx Buffer has its own Cancellation Finished Interrupt Enable bit.<br>0 Cancellation finished interrupt disabled<br>1 Cancellation finished interrupt enabled |

### 38.3.44 Tx Event FIFO Configuration Register (M\_CAN\_TXEFC)

Address: 0h base + F0h offset = F0h

|       |   |      |   |   |   |   |   |   |   |     |    |    |    |    |    |    |      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|------|---|---|---|---|---|---|---|-----|----|----|----|----|----|----|------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1    | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9   | 10 | 11 | 12 | 13 | 14 | 15 | 16   | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 | EFWM |   |   |   |   |   |   | 0 | EFS |    |    |    |    |    |    | EFSA |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0  |
| W     |   |      |   |   |   |   |   |   |   |     |    |    |    |    |    |    |      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0    | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

#### M\_CAN\_TXEFC field descriptions

| Field             | Description                                                                                                                                                                                                                                                                      |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–1<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                          |
| 2–7<br>EFWM       | Event FIFO Watermark<br><br><b>NOTE:</b> This field has Protected Write status.<br><br>0      Watermark interrupt disabled<br>1–32   Level for Tx Event FIFO watermark interrupt (IR[TEFW])<br>>32   Watermark interrupt disabled                                                |
| 8–9<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                          |
| 10–15<br>EFS      | Event FIFO Size<br><br>The Tx Event FIFO elements are indexed from 0 to EFS - 1<br><br><b>NOTE:</b> This field has Protected Write status.<br><br>0      Tx Event FIFO disabled<br>1–32   Number of Tx Event FIFO elements<br>>32   Values greater than 32 are interpreted as 32 |
| 16–29<br>EFSA     | Event FIFO Start Address<br><br>Start address of Tx Event FIFO in Message RAM (32-bit word address, <a href="#">Message RAM</a> ).<br><br><b>NOTE:</b> This field has Protected Write status.                                                                                    |
| 30–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                          |

### 38.3.45 Tx Event FIFO Status Register (M\_CAN\_TXEFS)

Address: 0h base + F4h offset = F4h

|       |   |   |   |   |   |   |      |     |   |   |    |      |    |    |    |    |  |
|-------|---|---|---|---|---|---|------|-----|---|---|----|------|----|----|----|----|--|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6    | 7   | 8 | 9 | 10 | 11   | 12 | 13 | 14 | 15 |  |
| R     | 0 |   |   |   |   |   | TEFL | EFF | 0 |   |    | EFPI |    |    |    |    |  |
| W     |   |   |   |   |   |   |      |     |   |   |    |      |    |    |    |    |  |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0    | 0   | 0 | 0 | 0  | 0    | 0  | 0  | 0  | 0  |  |

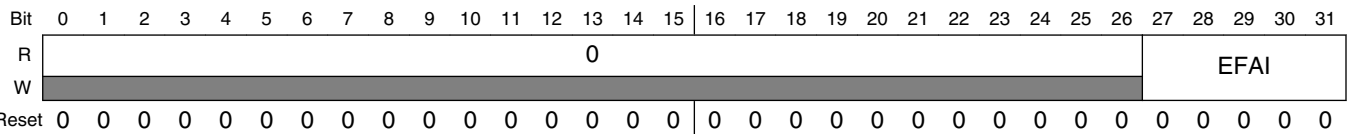
|       |    |    |    |      |    |    |    |    |    |    |    |      |    |    |    |    |  |
|-------|----|----|----|------|----|----|----|----|----|----|----|------|----|----|----|----|--|
| Bit   | 16 | 17 | 18 | 19   | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27   | 28 | 29 | 30 | 31 |  |
| R     | 0  |    |    | EFGI |    |    |    |    | 0  |    |    | EFFL |    |    |    |    |  |
| W     |    |    |    |      |    |    |    |    |    |    |    |      |    |    |    |    |  |
| Reset | 0  | 0  | 0  | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0  | 0  | 0  | 0  |  |

#### M\_CAN\_TXEFS field descriptions

| Field             | Description                                                                                                                                                                                                                                                         |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–5<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                             |
| 6<br>TEFL         | Tx Event FIFO Element Lost<br><br>This bit is a copy of interrupt flag IR[TEFL]. When IR[TEFL] is reset, this bit is also reset.<br><br>0 No Tx Event FIFO element lost<br>1 Tx Event FIFO element lost, also set after write attempt to Tx Event FIFO of size zero |
| 7<br>EFF          | Event FIFO Full<br><br>0 Tx Event FIFO not full<br>1 Tx Event FIFO full                                                                                                                                                                                             |
| 8–10<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                             |
| 11–15<br>EFPI     | Event FIFO Put Index<br><br>Tx Event FIFO write index pointer, range 0 to 31.                                                                                                                                                                                       |
| 16–18<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                             |
| 19–23<br>EFGI     | Event FIFO Get Index<br><br>Tx Event FIFO read index pointer, range 0 to 31.                                                                                                                                                                                        |
| 24–25<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                             |
| 26–31<br>EFFL     | Event FIFO Fill Level<br><br>Number of elements stored in Tx Event FIFO, range 0 to 32.                                                                                                                                                                             |

38.3.46 Tx Event FIFO Acknowledge Register (M\_CAN\_TXEFA)

Address: 0h base + F8h offset = F8h



M\_CAN\_TXEFA field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                               |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–26<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                   |
| 27–31<br>EFAI    | Event FIFO Acknowledge Index<br><br>After the CPU has read an element or a sequence of elements from the Tx Event FIFO it has to write the index of the last element read from Tx Event FIFO to EFAI. This will set the Tx Event FIFO Get Index TXEFS[EFGI] to EFAI + 1 and update the Event FIFO Fill Level TXEFS[EFFL]. |

38.4 Message RAM

For storage of Rx/Tx messages and for storage of the filter configuration a single- or dual-ported Message RAM has to be connected to the M\_CAN module.

NOTE

In case the MessageRAMis equipped with parity or ECC functionality, it is recommended to initialize the Message RAM after hardware reset by writing e.g. 0x00000000 to each Message RAM word to create valid parity/ECC checksums. This avoids that reading from uninitialized Message RAM sections will activate interrupt IR[BEC] (Bit Error Corrected) or IR[BEU] (Bit Error Uncorrected).

The Message RAM has a width of 32 bits. In case parity checking or ECC is used a respective number of bits has to be added to each word.

When operated in CAN FD mode the required Message RAM size strongly depends on the element size configured for Rx FIFO0, Rx FIFO1, Rx Buffers, and Tx Buffers via RXESC[F0DS], RXESC[F1DS], RXESC[RBDS], and TXESC[TBDS].

The M\_CAN module can be configured to allocate up to 4352 words in the Message RAM. For the actual amount of Message RAM on this chip, see the chip-specific M\_CAN information.

It is not necessary to configure each of the sections listed in the following figure, and there is no restriction with respect to the sequence of the sections.

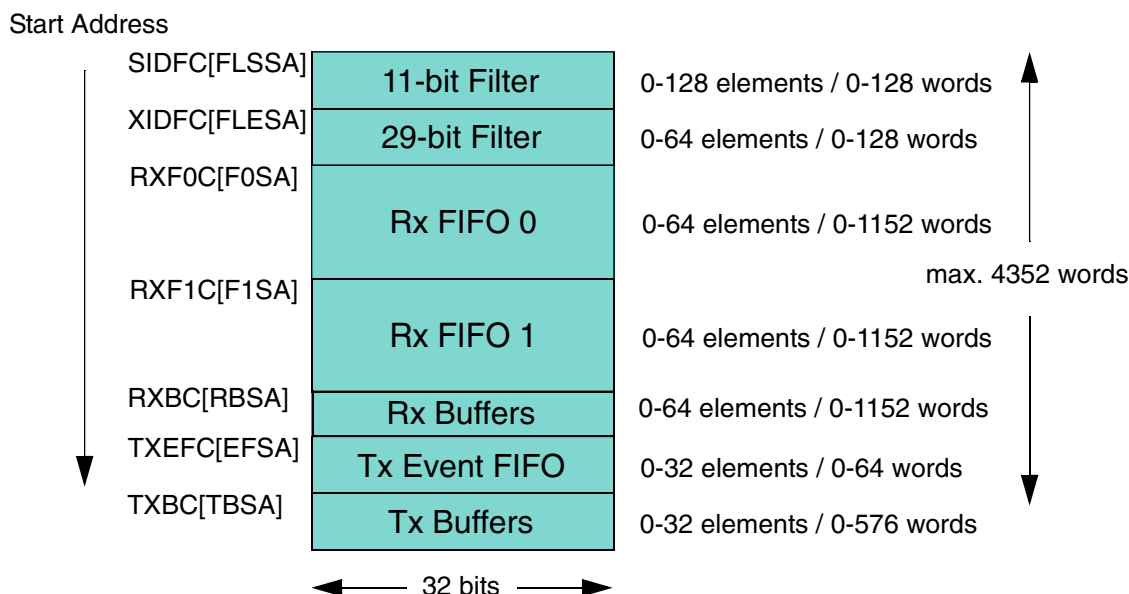


**Figure 38-49. Message RAM configuration**

When the M\_CAN addresses the Message RAM it addresses 32-bit words, not single bytes. The configurable start addresses are 32-bit word addresses i.e. only bits 15 to 2 are evaluated, the two least significant bits are ignored.

### NOTE

The M\_CAN does not check for erroneous configuration of the Message RAM. Especially the configuration of the start addresses of the different sections and the number of elements of each section has to be done carefully to avoid falsification or loss of data.

## 38.4.1 Rx Buffer and FIFO Element

Up to 64 Rx Buffers and two Rx FIFOs can be configured in the Message RAM. Each Rx FIFO section can be configured to store up to 64 received messages. The structure of a Rx Buffer / FIFO element is shown in the following table. The element size can be configured for storage of CAN FD messages with up to 64 bytes data field via register RXESC.

**Table 38-49. Rx Buffer and FIFO Element**

|    |             |             |             |          |            |             |             |          |            |            |  |  |            |  |  |  |   |   |  |  |  |   |
|----|-------------|-------------|-------------|----------|------------|-------------|-------------|----------|------------|------------|--|--|------------|--|--|--|---|---|--|--|--|---|
| 31 |             |             |             | 24       | 23         |             |             |          | 16         |            |  |  | 15         |  |  |  | 8 | 7 |  |  |  | 0 |
| R0 | E<br>S<br>I | X<br>T<br>D | R<br>T<br>R | ID[28:0] |            |             |             |          |            |            |  |  |            |  |  |  |   |   |  |  |  |   |
| R1 | ANMF        | FIDX[6:0]   |             |          | res        | F<br>D<br>F | B<br>R<br>S | DLC[3:0] |            | RXTS[15:0] |  |  |            |  |  |  |   |   |  |  |  |   |
| R2 | DB3[7:0]    |             |             |          | DB2[7:0]   |             |             |          | DB1[7:0]   |            |  |  | DB0[7:0]   |  |  |  |   |   |  |  |  |   |
| R3 | DB7[7:0]    |             |             |          | DB6[7:0]   |             |             |          | DB5[7:0]   |            |  |  | DB4[7:0]   |  |  |  |   |   |  |  |  |   |
|    | ...         |             |             |          | ...        |             |             |          | ...        |            |  |  | ...        |  |  |  |   |   |  |  |  |   |
| Rn | DBm[7:0]    |             |             |          | DBm-1[7:0] |             |             |          | DBm-2[7:0] |            |  |  | DBm-3[7:0] |  |  |  |   |   |  |  |  |   |

**Table 38-50. Rx Buffer and FIFO Element Descriptions**

|                      |                                                                                                                                                                                                                                                                                                                                                                                              |
|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>R0 Bit 31</b>     | ESI: Error State Indicator<br>0 Transmitting node is error active<br>1 Transmitting node is error passive                                                                                                                                                                                                                                                                                    |
| <b>R0 Bit 30</b>     | XTD: Extended Identifier<br>Signals to the Host whether the received frame has a standard or extended identifier.<br>0 11-bit standard identifier<br>1 29-bit standard identifier                                                                                                                                                                                                            |
| <b>R0 Bit 29</b>     | RTR: Remote Transmission Request<br>Signals to the Host whether the received frame is a data frame or a remote frame.<br>0 Received frame is a data frame<br>1 Received frame is a remote frame<br><b>NOTE:</b> There are no remote frames in CAN FD format. In CAN FD frames (FDF = 1'), the dominant RRS (Remote Request Substitution) bit replaces bit RTR (Remote Transmission Request). |
| <b>R0 Bits 28:0</b>  | ID[28:0]: Identifier<br>Standard or extended identifier depending on bit XTD. A standard identifier is stored into ID[28:18].                                                                                                                                                                                                                                                                |
| <b>R1 Bit 31</b>     | ANMF: Accepted Non-matching Frame<br>Acceptance of non-matching frames may be enabled via GFC[ANFS] and GFC[ANFE].<br>0 Received frame matching filter index FIDX<br>1 Received frame did not match any Rx filter element                                                                                                                                                                    |
| <b>R1 Bits 30:24</b> | FIDX[6:0]: Filter Index<br>0-127 Index of matching Rx acceptance filter element (invalid if ANMF = 1). Range is 0 to SIDFC[LSS] - 1 resp. XIDFC[LSE] - 1.                                                                                                                                                                                                                                    |
| <b>R1 Bit 21</b>     | FDF: FD Format<br>0 Standard frame format<br>1 CAN FD frame format (new DLC-coding and CRC)                                                                                                                                                                                                                                                                                                  |
| <b>R1 Bit 20</b>     | BRS: Bit Rate Switch                                                                                                                                                                                                                                                                                                                                                                         |

*Table continues on the next page...*

**Table 38-50. Rx Buffer and FIFO Element Descriptions (continued)**

|                      |                                                                                                                                                                                                   |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                      | 0 Frame received without bit rate switching<br>1 Frame received with bit rate switching                                                                                                           |
| <b>R1 Bits 19:16</b> | DLC[3:0]: Data Length Code<br>0-8 CAN + CAN FD: received frame has 0-8 data bytes<br>9-15 CAN: received frame has 8 data bytes<br>9-15 CAN FD: received frame has 12/16/20/24/32/48/64 data bytes |
| <b>R1 Bits 15:0</b>  | RXTS[15:0]: Rx Timestamp<br>Timestamp Counter value captured on start of frame reception. Resolution depending on configuration of the Timestamp Counter Prescaler TSCC[TCP].                     |
| <b>R2 Bits 31:24</b> | DB3[7:0]: Data Byte 3                                                                                                                                                                             |
| <b>R2 Bits 23:16</b> | DB2[7:0]: Data Byte 2                                                                                                                                                                             |
| <b>R2 Bits 15:8</b>  | DB1[7:0]: Data Byte 1                                                                                                                                                                             |
| <b>R2 Bits 7:0</b>   | DB0[7:0]: Data Byte 0                                                                                                                                                                             |
| <b>R3 Bits 31:24</b> | DB7[7:0]: Data Byte 7                                                                                                                                                                             |
| <b>R3 Bits 23:16</b> | DB6[7:0]: Data Byte 6                                                                                                                                                                             |
| <b>R3 Bits 15:8</b>  | DB5[7:0]: Data Byte 5                                                                                                                                                                             |
| <b>R3 Bits 7:0</b>   | DB4[7:0]: Data Byte 4                                                                                                                                                                             |
| ...                  | ...                                                                                                                                                                                               |
| <b>Rn Bits 31:24</b> | DBm[7:0]: Data Byte m                                                                                                                                                                             |
| <b>Rn Bits 23:16</b> | DBm-1[7:0]: Data Byte m-1                                                                                                                                                                         |
| <b>Rn Bits 15:8</b>  | DBm-2[7:0]: Data Byte m-2                                                                                                                                                                         |
| <b>Rn Bits 7:0</b>   | DBm-3[7:0]: Data Byte m-3                                                                                                                                                                         |

**NOTE**

Depending on the configuration of the element size (RXESC), between two and sixteen 32-bit words ( $R_n = 3 \dots 17$ ) are used for storage of a CAN message's data field.

**38.4.2 Tx Buffer Element**

The Tx Buffers section can be configured to hold dedicated Tx Buffers as well as a Tx FIFO / Tx Queue. In case that the Tx Buffers section is shared by dedicated Tx buffers and a Tx FIFO / Tx Queue, the dedicated Tx Buffers start at the beginning of the Tx Buffers section followed by the buffers assigned to the Tx FIFO or Tx Queue. The Tx Handler distinguishes between dedicated Tx Buffers and Tx FIFO / Tx Queue by evaluating the Tx Buffer configuration TXBC.TFQS and TXBC.NDTB. The element size can be configured for storage of CAN FD messages with up to 64 bytes data field via register TXESC.

Table 38-51. Tx Buffer Element

|     |          |     |     |            |     |      |            |     |
|-----|----------|-----|-----|------------|-----|------|------------|-----|
|     | 31       | 24  | 23  | 16         | 15  | 8    | 7          | 0   |
| T0  | ESI      | XTD | RTR | ID[28:0]   |     |      |            |     |
| T1  | MM[7:0]  |     |     | EFC        | res | FDFS | DLC[3:0]   | res |
| T2  | DB3[7:0] |     |     | DB2[7:0]   |     |      | DB1[7:0]   |     |
| T3  | DB7[7:0] |     |     | DB6[7:0]   |     |      | DB5[7:0]   |     |
| ... | ...      |     |     | ...        |     |      | ...        |     |
| Tn  | DBm[7:0] |     |     | DBm-1[7:0] |     |      | DBm-2[7:0] |     |
|     |          |     |     |            |     |      | DBm-3[7:0] |     |

Table 38-52. Tx Buffer Element Description

|                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>T0 Bit 31</b>     | <p>ESI: Error State Indicator</p> <p>0 ESI bit in CAN FD format depends only on error passive flag</p> <p>1 ESI bit in CAN FD format transmitted recessive</p> <p><b>NOTE:</b> The ESI bit of the transmit buffer is OR-ed with the error passive flag to decide the value of the ESI bit in the transmitted FD frame. As required by the CAN FD protocol specification, an error active node may optionally transmit the ESI bit recessive, but an error passive node will always transmit the ESI bit recessive.</p> |
| <b>T0 Bit 30</b>     | <p>XTD: Extended Identifier</p> <p>0 11-bit standard identifier</p> <p>1 29-bit extended identifier</p>                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>T0 Bit 29</b>     | <p>RTR: Remote Transmission Request</p> <p>0 Transmit data frame</p> <p>1 Transmit remote frame</p> <p><b>NOTE:</b> When RTR = 1, the M_CAN transmits a remote frame according to ISO11898-1, even if CCCR[FDOE] enables the transmission in CAN FD format.</p>                                                                                                                                                                                                                                                        |
| <b>T0 Bit 28:0</b>   | <p>ID[28:0]: Identifier</p> <p>Standard or extended identifier depending on bit XTD. A standard identifier has to be written to ID[28:18].</p>                                                                                                                                                                                                                                                                                                                                                                         |
| <b>T1 Bits 31:24</b> | <p>MM[7:0]: Message Marker</p> <p>Written by CPU during Tx Buffer configuration. Copied into Tx Event FIFO element for identification of Tx message status.</p>                                                                                                                                                                                                                                                                                                                                                        |
| <b>T1 Bit 23</b>     | <p>EFC: Event FIFO Control</p> <p>0 Do not store Tx events</p> <p>1 Store Tx events</p>                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>T1 Bit 21</b>     | <p>FDF: FD Format</p> <p>0 Frame transmitted in Classic CAN format</p> <p>1 Frame transmitted in CAN FD format</p>                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>T1 Bit 20</b>     | <p>BRS: Bit Rate Switching</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

Table continues on the next page...



**Table 38-52. Tx Buffer Element Description (continued)**

|                      |                                                                                                                                                                                                                                                                                     |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                      | 0 CAN FD frames transmitted without bit rate switching<br>1 CAN FD frames transmitted with bit rate switching<br><b>NOTE:</b> Bits ESL, FDF, and BRS are only evaluated when CAN FD operation is enabled CCCR[FDOE] = 1. Bit BRS is only evaluated when in addition CCCR[BRSE] = 1. |
| <b>T1 Bits 19:16</b> | DLC[3:0]: Data Length Code<br>0-8 CAN + CAN FD: Transmit frame has 0-8 data bytes<br>9-15 CAN: Transmit frame has 8 data bytes<br>9-15 CAN FD: Transmit frame has 12/16/20/24/32/48/64 data bytes                                                                                   |
| <b>T2 Bits 31:24</b> | DB3[7:0]: Data Byte 3                                                                                                                                                                                                                                                               |
| <b>T2 Bits 23:16</b> | DB2[7:0]: Data Byte 2                                                                                                                                                                                                                                                               |
| <b>T2 Bits 15:8</b>  | DB1[7:0]: Data Byte 1                                                                                                                                                                                                                                                               |
| <b>T2 Bits 7:0</b>   | DB0[7:0]: Data Byte 0                                                                                                                                                                                                                                                               |
| <b>T3 Bits 31:24</b> | DB7[7:0]: Data Byte 7                                                                                                                                                                                                                                                               |
| <b>T3 Bits 23:16</b> | DB6[7:0]: Data Byte 6                                                                                                                                                                                                                                                               |
| <b>T3 Bits 15:8</b>  | DB5[7:0]: Data Byte 5                                                                                                                                                                                                                                                               |
| <b>T3 Bits 7:0</b>   | DB4[7:0]: Data Byte 4                                                                                                                                                                                                                                                               |
| ...                  | ...                                                                                                                                                                                                                                                                                 |
| <b>Tn Bits 31:24</b> | DBm[7:0]: Data Byte m                                                                                                                                                                                                                                                               |
| <b>Tn Bits 23:16</b> | DBm-1[7:0]: Data Byte m-1                                                                                                                                                                                                                                                           |
| <b>Tn Bits 15:8</b>  | DBm-2[7:0]: Data Byte m-2                                                                                                                                                                                                                                                           |
| <b>Tn Bits 7:0</b>   | DBm-3[7:0]: Data Byte m-3                                                                                                                                                                                                                                                           |

**NOTE**

Depending on the configuration of the element size (TXESC), between two and sixteen 32-bit words (Tn = 3 ..17) are used for storage of a CAN message's data field.

**38.4.3 Tx Event FIFO Element**

Each element stores information about transmitted messages. By reading the Tx Event FIFO the Host CPU gets this information in the order the messages were transmitted. Status information about the Tx Event FIFO can be obtained from register TXEFS.

**Table 38-53. Tx Event FIFO Element**

|    |         |             |             |          |             |             |             |          |    |            |  |  |  |   |   |   |  |  |  |
|----|---------|-------------|-------------|----------|-------------|-------------|-------------|----------|----|------------|--|--|--|---|---|---|--|--|--|
| 31 |         |             |             | 24       | 23          |             |             |          | 16 | 15         |  |  |  | 8 | 7 | 0 |  |  |  |
| E0 | E<br>SI | X<br>T<br>D | R<br>T<br>R | ID[28:0] |             |             |             |          |    |            |  |  |  |   |   |   |  |  |  |
| E1 | MM[7:0] |             |             |          | ET[1:<br>0] | F<br>D<br>F | B<br>R<br>S | DLC[3:0] |    | TXTS[15:0] |  |  |  |   |   |   |  |  |  |

**Table 38-54. Tx Event FIFO Element Description**

|                      |                                                                                                                                                                                                               |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>E0 Bit 31</b>     | ESI: Error State Indicator<br>0 Transmitting node is error active<br>1 Transmitting node is error passive                                                                                                     |
| <b>E0 Bit 30</b>     | XTD: Extended Identifier<br>0 11-bit standard identifier<br>1 29-bit extended identifier                                                                                                                      |
| <b>E0 Bit 29</b>     | RTR: Remote Transmission Request<br>0 Data frame transmitted<br>1 Remote frame transmitted                                                                                                                    |
| <b>E0 Bits 28:0</b>  | ID[28:0]: Identifier<br>Standard or extended identifier depending on bit XTD. A standard identifier is stored into ID[28:18].                                                                                 |
| <b>E1 Bits 31:24</b> | MM[7:0]: Message Marker<br>Copied from Tx Buffer into Tx Event FIFO element for identification of Tx message status.                                                                                          |
| <b>E1 Bit 23:22</b>  | ET[1:0]: Event Type<br>00 Reserved<br>01 Tx event<br>10 Transmission in spite of cancellation (always set for transmissions in DAR mode)<br>11 Reserved                                                       |
| <b>E1 Bit 21</b>     | FDF: FD Format<br>0 Standard frame format<br>1 CAN FD frame format (new DLC-coding and CRC)                                                                                                                   |
| <b>E1 Bit 20</b>     | BRS: Bit Rate Switch<br>0 Frame transmitted without bit rate switching<br>1 Frame transmitted with bit rate switching                                                                                         |
| <b>E1 Bits 19:16</b> | DLC[3:0]: Data Length Code<br>0-8 CAN + CAN FD: Frame with 0-8 data bytes transmitted<br>9-15 CAN: Frame with 8 data bytes transmitted<br>9-15 CAN FD: frame with 12/16/20/24/32/48/64 data bytes transmitted |
| <b>E1 Bits 15:0</b>  | TXTS[15:0]: Tx Timestamp<br>Timestamp Counter value captured on start of frame transmission. Resolution depending on configuration of the Timestamp Counter Prescaler TSCC.TCP.                               |

### 38.4.4 Standard Message ID Filter Element

Up to 128 filter elements can be configured for 11-bit standard IDs. When accessing a Standard Message ID Filter element, its address is the Filter List Standard Start Address SIDFC[FLSSA] plus the index of the filter element (0...127).

**Table 38-55. Standard Message ID Filter Element**

|    |          |           |             |    |     |   |             |   |
|----|----------|-----------|-------------|----|-----|---|-------------|---|
|    | 31       | 24        | 23          | 16 | 15  | 8 | 7           | 0 |
| S0 | SFT[1:0] | SFEC[2:0] | SFID1[10:0] |    | res |   | SFID2[10:0] |   |

**Table 38-56. Standard Message ID Filter Element Field Description**

|                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>S0 Bits 31:30</b> | <p>SFT[1:0]: Standard Filter Type</p> <p>00 Range filter from SFID1 to SFID2 (SFID2 &gt;= SFID1)</p> <p>01 Dual ID filter for SFID1 or SFID2</p> <p>10 Classic filter: SFID1 = filter, SFID2 = mask</p> <p>11 Filter element disabled</p> <p><b>NOTE:</b> With SFT = 11 the filter element is disabled and the acceptance filtering continues (same behavior as with SFEC = 000).</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>S0 Bits 29:27</b> | <p>SFEC[2:0]: Standard Filter Element Configuration</p> <p>All enabled filter elements are used for acceptance filtering of standard frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If SFEC = 100, 101, or 110 a match sets interrupt flag IR[HPM] and, if enabled, an interrupt is generated. In this case register HPMS is updated with the status of the priority match.</p> <p>000 Disable filter element</p> <p>001 Store in Rx FIFO 0 if filter matches</p> <p>010 Store in Rx FIFO 1 if filter matches</p> <p>011 Reject ID if filter matches</p> <p>100 Set priority if filter matches</p> <p>101 Set priority and store in FIFO 0 if filter matches</p> <p>110 Set priority and store in FIFO 1 if filter matches</p> <p>111 Store into Rx Buffer or as debug message, configuration of SFT[1:0] ignored</p> |
| <b>S0 Bits 26:16</b> | <p>SFID1[10:0]: Standard Filter ID 1</p> <p>First ID of standard ID filter element.</p> <p>When filtering for Rx Buffers or for debug messages this field defines the ID of a standard message to be stored. The received identifiers must match exactly, no masking mechanism is used.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>S0 Bits 10:0</b>  | <p>SFID2[10:0]: Standard Filter ID 2</p> <p>Overall, this bit field has a different meaning depending on the configuration of SFEC:</p> <p>SFEC = 001...110 Second ID of standard ID filter element</p> <p>SFEC = 111 Filter for Rx Buffers or for debug messages</p> <p>SFID2[10:9]: Decides whether the received message is stored into an Rx Buffer or treated as message A, B, or C of the debug message sequence.</p> <p>00 Store message into an Rx Buffer</p> <p>01 Debug Message A</p> <p>10 Debug Message B</p> <p>11 Debug Message C</p>                                                                                                                                                                                                                                                                                                                                                    |

*Table continues on the next page...*

**Table 38-56. Standard Message ID Filter Element Field Description (continued)**

|  |                                                                                                                                                                                                                                                                  |
|--|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | SFID2[8:6]: Is used to control the M_CAN filter event pins at the Extension Interface. A one at the respective bit position enables generation of a pulse at the related filter event pin with the duration of one Host clock period in case the filter matches. |
|  | SFID2[5:0]: Defines the offset to the Rx Buffer Start Address RXBC.RBSA for storage of a matching message.                                                                                                                                                       |

### 38.4.5 Extended Message ID Filter Element

Up to 64 filter elements can be configured for 29-bit extended IDs. When accessing an Extended Message ID Filter element, its address is the Filter List Extended Start Address XIDFC[FLESA] plus two times the index of the filter element (0...63).

**Table 38-57. Extended Message ID Filter Element**

|    |           |             |             |    |    |   |   |   |
|----|-----------|-------------|-------------|----|----|---|---|---|
|    | 31        | 24          | 23          | 16 | 15 | 8 | 7 | 0 |
| F0 | EFEC[2:0] | EFID1[28:0] |             |    |    |   |   |   |
| F1 | EFT[1:0]  | res         | EFID2[28:0] |    |    |   |   |   |

**Table 38-58. Extended Message ID Filter Element Field Description**

|               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| F0 Bits 31:29 | <p>EFEC[2:0]: Extended Filter Element Configuration</p> <p>All enabled filter elements are used for acceptance filtering of extended frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If EFEC = 100, 101, or 110 a match sets interrupt flag IR[HPM] and, if enabled, an interrupt is generated. In this case register HPMS is updated with the status of the priority match.</p> <p>000 Disable filter element</p> <p>001 Store in Rx FIFO 0 if filter matches</p> <p>010 Store in Rx FIFO 1 if filter matches</p> <p>011 Reject ID if filter matches</p> <p>100 Set priority if filter matches</p> <p>101 Set priority and store in FIFO 0 if filter matches</p> <p>110 Set priority and store in FIFO 1 if filter matches</p> <p>111 Store into Rx Buffer or as debug message, configuration of EFT[1:0] ignored</p> |
| F0 Bits 28:0  | <p>EFID1[28:0]: Extended Filter ID 1</p> <p>First ID of extended ID filter element.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |

*Table continues on the next page...*

**Table 38-58. Extended Message ID Filter Element Field Description (continued)**

|                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                      | When filtering for Rx Buffers or for debug messages this field defines the ID of an extended message to be stored. The received identifiers must match exactly, only XIDAM masking mechanism (see <a href="#">Extended Message ID Filtering</a> ) is used.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>F1 Bits 31:30</b> | <p>EFT[1:0]: Extended Filter Type</p> <p>00 Range filter from EFID1 to EFID2 (EFID2 &gt;= EFID1)</p> <p>01 Dual ID filter for EFID1 or EFID2</p> <p>10 Classic filter: EFID1 = filter, EFID2 = mask</p> <p>11 Range filter from EFID1 to EFID2 (EFID2 &gt;= EFID1), XIDAM mask not applied</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>F1 Bits 28:0</b>  | <p>EFID2[28:0]: Extended Filter ID 2</p> <p>Overall, this bit field has a different meaning depending on the configuration of EFEC:</p> <p>EFEC = 001...110 Second ID of extended ID filter element</p> <p>EFEC = 111 Filter for Rx Buffers or for debug messages</p> <p>EFID2[10:9]: Decides whether the received message is stored into an Rx Buffer or treated as message A, B, or C of the debug message sequence.</p> <p>00 Store message into an Rx Buffer</p> <p>01 Debug Message A</p> <p>10 Debug Message B</p> <p>11 Debug Message C</p> <p>EFID2[8:6]: Is used to control the filter event pins at the Extension Interface. A one at the respective bit position enables generation of a pulse at the related filter event pin with the duration of one Host clock period in case the filter matches.</p> <p>EFID2[5:0]: Defines the offset to the Rx Buffer Start Address RXBC[RBSA] for storage of a matching message.</p> |

## 38.5 Functional Description

### 38.5.1 Operating Modes

#### 38.5.1.1 Software Initialization

Software initialization is started by setting bit CCCR[INIT], either by software or by a hardware reset, when an uncorrected bit error was detected in the Message RAM, or by going Bus\_Off. While CCCR[INIT] is set, message transfer from and to the CAN bus is stopped, the status of the CAN bus output M\_CAN\_Tx is recessive (HIGH). The counters of the Error Management Logic EML are unchanged. Setting CCCR[INIT] does not change any configuration register. Resetting CCCR[INIT] finishes the software

initialization. Afterwards the Bit Stream Processor BSP synchronizes itself to the data transfer on the CAN bus by waiting for the occurrence of a sequence of 11 consecutive recessive bits (= Bus\_Idle) before it can take part in bus activities and start the message transfer.

Access to the M\_CAN configuration registers is only enabled when both bits CCCR[INIT] and CCCR[CCE] are set (protected write).

CCCR[CCE] can only be set/reset while CCCR[INIT] = 1. CCCR[CCE] is automatically reset when CCCR[INIT] is reset.

The following registers are reset when CCCR[CCE] is set

- HPMS – High Priority Message Status
- RXF0S – Rx FIFO 0 Status
- RXF1S – Rx FIFO 1 Status
- TXFQS – Tx FIFO/Queue Status
- TXBRP – Tx Buffer Request Pending
- TXBTO – Tx Buffer Transmission Occurred
- TXBCF – Tx Buffer Cancellation Finished
- TXEFS – Tx Event FIFO Status

The Timeout Counter value TOCV[TOC] is preset to the value configured by TOCC[TOP] when CCCR[CCE] is set.

In addition the state machines of the Tx Handler and Rx Handler are held in idle state while CCCR[CCE] = 1.

The following registers are only writable while CCCR[CCE] = 0

- TXBAR – Tx Buffer Add Request
- TXBCR – Tx Buffer Cancellation Request

CCCR[TEST] and CCCR[MON] can only be set by the Host while CCCR[INIT] = 1 and CCCR[CCE] = 1. Both bits may be reset at any time. CCCR[DAR] can only be set/reset while CCCR[INIT] = 1 and CCCR[CCE] = 1.

### **NOTE**

In case the Message RAM is equipped with parity or ECC functionality, it is recommended to initialize the Message RAM after hardware reset by writing e.g. 0x00000000 to each

Message RAM word to create valid parity/ECC checksums. This avoids that reading from uninitialized Message RAM sections will activate interrupt IR.BEC (Bit Error Corrected) or IR.BEU (Bit Error Uncorrected).

### 38.5.1.2 Normal Operation

Once the M\_CAN is initialized and CCCR.INIT is reset to zero, the M\_CAN synchronizes itself to the CAN bus and is ready for communication.

After passing the acceptance filtering, received messages including Message ID and DLC are stored into a dedicated Rx Buffer or into Rx FIFO 0 or Rx FIFO 1.

For messages to be transmitted dedicated Tx Buffers and/or a Tx FIFO or a Tx Queue can be initialized or updated. Automated transmission on reception of remote frames is not implemented.

### 38.5.1.3 CAN FD Operation

There are two variants in the CAN FD frame transmission, first the CAN FD frame without bit rate switching. The second variant is the CAN FD frame where control field, data field, and CRC field are transmitted with a higher bit rate than the beginning and the end of the frame.

The previously reserved bit in CAN frames with 11-bit identifiers and the first previously reserved bit in CAN frames with 29-bit identifiers will now be decoded as FDF bit. FDF = recessive signifies a CAN FD frame, FDF = dominant signifies a Classic CAN frame. In a CAN FD frame, the two bits following FDF, res and BRS, decide whether the bit rate inside of this CAN FD frame is switched. A CAN FD bit rate switch is signified by res = dominant and BRS = recessive. The coding of res = recessive is reserved for future expansion of the protocol. In case the M\_CAN receives a frame with FDF = recessive and res = recessive, it will signal a Protocol Exception Event by setting bit PSR[PXE]. When Protocol Exception Handling is enabled (CCCR[PXHD] = 0), this causes the operation state to change from Receiver (PSR[ACT] = 10) to Integrating (PSR[ACT] = 00) at the next sample point. In case Protocol Exception Handling is disabled (CCCR[PXHD] = 1), the M\_CAN will treat a recessive res bit as an form error and will respond with an error frame.

CAN FD operation is enabled by programming CCCR[FDOE]. In case CCCR[FDOE] = 1, transmission and reception of CAN FD frames is enabled. Transmission and reception of Classic CAN frames is always possible. Whether a CAN FD frame or a Classic CAN frame is transmitted can be configured via bit FDF in the respective Tx Buffer element.

With  $\text{CCCR}[\text{FDOE}] = 0$ , received frames are interpreted as Classic CAN frames, which leads to the transmission of an error frame when receiving a CAN FD frame. When CAN FD operation is disabled, no CAN FD frames are transmitted even if bit FDF of a Tx Buffer element is set.  $\text{CCCR}[\text{FDOE}]$  and  $\text{CCCR}[\text{BRSE}]$  can only be changed while  $\text{CCCR}[\text{INIT}]$  and  $\text{CCCR}[\text{CCE}]$  are both set.

With  $\text{CCCR}[\text{FDOE}] = 0$ , the setting of bits FDF and BRS is ignored and frames are transmitted in Classic CAN format. With  $\text{CCCR}[\text{FDOE}] = 1$  and  $\text{CCCR}[\text{BRSE}] = 0$ , only bit FDF of a Tx Buffer element is evaluated. With  $\text{CCCR}[\text{FDOE}] = 1$  and  $\text{CCCR}[\text{BRSE}] = 1$ , transmission of CAN FD frames with bit rate switching is enabled. All Tx Buffer elements with bits FDF and BRS set are transmitted in CAN FD format with bit rate switching.

A mode change during CAN operation is only recommended under the following conditions:

- The failure rate in the CAN FD data phase is significant higher than in the CAN FD arbitration phase. In this case disable the CAN FD bit rate switching option for transmissions.
- During system startup all nodes are transmitting Classic CAN messages until it is verified that they are able to communicate in CAN FD format. If this is true, all nodes switch to CAN FD operation.
- Wake-up messages in CAN Partial Networking have to be transmitted in Classic CAN format.
- End-of-line programming in case not all nodes are CAN FD capable. Non CAN FD nodes are held in Silent mode until programming has completed. Then all nodes switch back to Classic CAN communication.

In the CAN FD format, the coding of the DLC differs from the standard CAN format. The DLC codes 0 to 8 have the same coding as in standard CAN, the codes 9 to 15, which in standard CAN all code a data field of 8 bytes, are coded according to the following table.

**Table 38-59. Coding of DLC in CAN FD**

| DLC                  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
|----------------------|----|----|----|----|----|----|----|
| Number of data bytes | 12 | 16 | 20 | 24 | 32 | 48 | 64 |

In CAN FD frames, the bit timing will be switched inside the frame, after the BRS (Bit Rate Switch) bit, if this bit is recessive. Before the BRS bit, in the CAN FD arbitration phase, the standard CAN bit timing is used as defined by the Bit Timing and Prescaler Register BTP. In the following CAN FD data phase, the fast CAN bit timing is used as



defined by the Fast Bit Timing and Prescaler Register FBTP. The bit timing is switched back from the fast timing at the CRC delimiter or when an error is detected, whichever occurs first.

The maximum configurable bit rate in the CAN FD data phase depends on the CAN clock frequency. Example: with a CAN clock frequency of 20 MHz and the shortest configurable bit time of 4 tq, the bit rate in the data phase is 5 Mbit/s.

In both data frame formats, CAN FD long and CAN FD fast, the value of the bit ESI (Error Status Indicator) is determined by the transmitter's error state at the start of the transmission. If the transmitter is error passive, ESI is transmitted recessive, else it is transmitted dominant.

### 38.5.1.4 Transmitter Delay Compensation

During the data phase of a CAN FD transmission only one node is transmitting, all others are receivers. The length of the bus line has no impact. When transmitting via pin M\_CAN\_Tx the M\_CAN receives the transmitted data from its local CAN transceiver via M\_CAN\_Rx pin. The received data is delayed by the transmitter delay. In case this delay is greater than TSEG1 (time segment before sample point), a bit error is detected. In order to enable a data phase bit time that is even shorter than the transmitter delay, the delay compensation is introduced. Without transmitter delay compensation, the bit rate in the data phase of a CAN FD frame is limited by the transmitter delay.

#### 38.5.1.4.1 Description

The M\_CAN's protocol unit has implemented a delay compensation mechanism to compensate the transmitter delay, thereby enabling transmission with higher bit rates during the CAN FD data phase independent of the delay of a specific CAN transceiver.

To check for bit errors during the data phase of transmitting nodes, the delayed transmit data is compared against the received data at the Secondary Sample Point SSP. If a bit error is detected, the transmitter will react on this bit error at the next following regular sample point. During arbitration phase the delay compensation is always disabled.

The transmitter delay compensation enables configurations where the data bit time is shorter than the transmitter delay, it is described in detail in the new ISO11898-1. It is enabled by setting bit DBTP[TDC].

The received bit is compared against the transmitted bit at the SSP. The SSP position is defined as the sum of the measured delay from the M\_CAN's transmit output M\_CAN\_Tx through the transceiver to the receive input M\_CAN\_Rx plus the transmitter delay compensation offset as configured by TDCR[TDCO]. The transmitter delay

compensation offset is used to adjust the position of the SSP inside the received bit (e.g. half of the bit time in the data phase). The position of the secondary sample point is rounded down to the next integer number of mtq.

PSR[TDCV] shows the actual transmitter delay compensation value. PSR[TDCV] is cleared when CCCR[INIT] is set and is updated at each transmission of an FD frame while DBTP[TDC] is set.

The following boundary conditions have to be considered for the transmitter delay compensation implemented in the M\_CAN:

- The sum of the measured delay from M\_CAN\_Tx to M\_CAN\_Rx and the configured transmitter delay compensation offset TDCR[TDCO] has to be less than 6 bit times in the data phase.
- The sum of the measured delay from M\_CAN\_Tx to M\_CAN\_Rx and the configured transmitter delay compensation offset TDCR[TDCO] has to be less or equal 127 mtq. In case this sum exceeds 127 mtq, the maximum value of 127 mtq is used for transmitter delay compensation.
- The data phase ends at the sample point of the CRC delimiter, that stops checking of receive bits at the SSPs.

#### 38.5.1.4.2 Transmitter delay compensation measurement

If transmitter delay compensation is enabled by programming DBTP[TDC] = 1, the measurement is started within each transmitted CAN FD frame at the falling edge of bit FDF to bit res. The measurement is stopped when this edge is seen at the receive input M\_CAN\_Rx of the transmitter. The resolution of this measurement is one mtq.

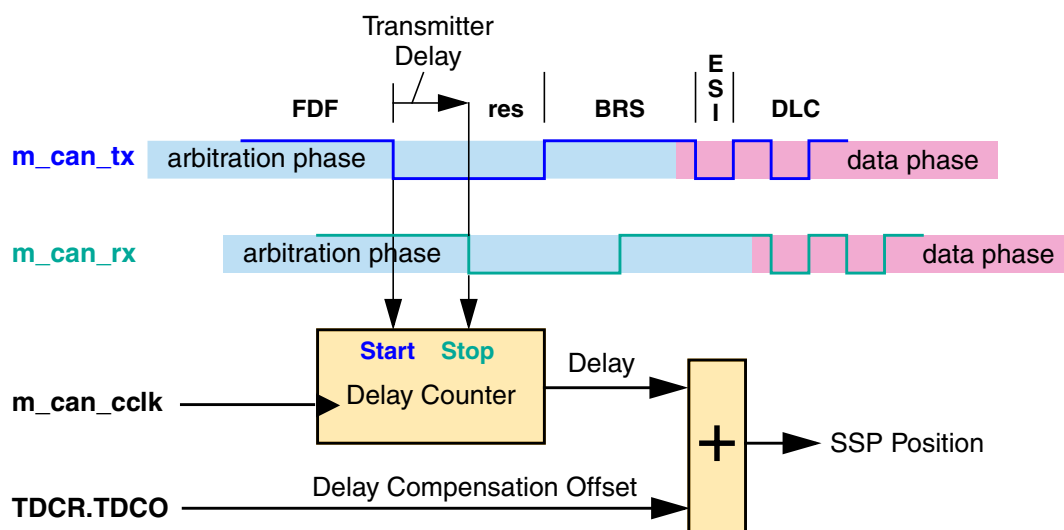


Figure 38-50. Transmitter delay measurement

To avoid that a dominant glitch inside the received FDF bit ends the delay compensation measurement before the falling edge of the received res bit, resulting in a too early SSP position, the use of a transmitter delay compensation filter window can be enabled by programming TDCR[TDCF]. This defines a minimum value for the SSP position. Dominant edges on M\_CAN\_Rx, that would result in an earlier SSP position are ignored for transmitter delay measurement. The measurement is stopped when the SSP position is at least TDCR[TDCF] AND M\_CAN\_Rx is low.

### 38.5.1.5 Restricted Operation Mode

In Restricted Operation Mode the node is able to receive data and remote frames and to give acknowledge to valid frames, but it does not send data frames, remote frames, active error frames, or overload frames. In case of an error condition or overload condition, it does not send dominant bits, instead it waits for the occurrence of bus idle condition to resynchronize itself to the CAN communication. The error counters (ECR[REC], ECR[TEC]) are frozen while Error Logging (ECR[CEL]) is active. The Host can set the M\_CAN into Restricted Operation mode by setting bit CCCR[ASM]. The bit can only be set by the Host when both CCCR[CCE] and CCCR[INIT] are set to 1. The bit can be reset by the Host at any time.

Restricted Operation Mode is automatically entered when the Tx Handler was not able to read data from the Message RAM in time. To leave Restricted Operation Mode, the Host CPU has to reset CCCR[ASM].

The Restricted Operation Mode can be used in applications that adapt themselves to different CAN bit rates. In this case the application tests different bit rates and leaves the Restricted Operation Mode after it has received a valid frame.

#### NOTE

The Restricted Operation Mode must not be combined with the Loop Back Mode (internal or external).

### 38.5.1.6 Bus Monitoring Mode

The M\_CAN is set in Bus Monitoring Mode by programming CCCR.MON to one. In Bus Monitoring Mode (see ISO11898-1, 10.12 Bus monitoring), the M\_CAN is able to receive valid data frames and valid remote frames, but cannot start a transmission. In this mode, it sends only recessive bits on the CAN bus. If the M\_CAN is required to send a dominant bit (ACK bit, overload flag, active error flag), the bit is rerouted internally so that the M\_CAN monitors this dominant bit, although the CAN bus may remain in recessive state. In Bus Monitoring Mode, register TXBRP is held in reset state.

The Bus Monitoring Mode can be used to analyze the traffic on a CAN bus without affecting it by the transmission of dominant bits. The following figure shows the connection of signals M\_CAN Tx and Rx to the M\_CAN in Bus Monitoring Mode.

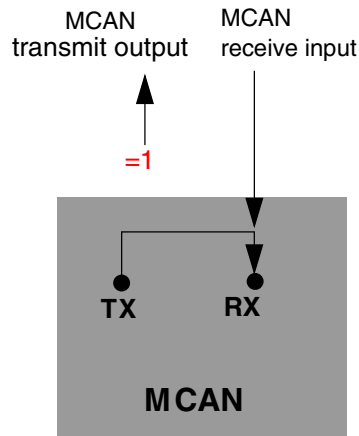


**Figure 38-51. Pin Control in Bus Monitoring Mode**

### 38.5.1.7 Disabled Automatic Retransmission

According to the CAN Specification (see ISO11898-1, 6.3.3 Recovery Management), the M\_CAN provides means for automatic retransmission of frames that have lost arbitration or that have been disturbed by errors during transmission. By default automatic retransmission is enabled. To support time-triggered communication as described in ISO 11898-1, chapter 9.2, the automatic retransmission may be disabled via CCCR[DAR].

#### 38.5.1.7.1 Frame Transmission in DAR mode

In DAR mode all transmissions are automatically cancelled after they started on the CAN bus. A Tx Buffer's Tx Request Pending bit TXBRP[TRPx] is reset after successful transmission, when a transmission has not yet been started at the point of cancellation, has been aborted due to lost arbitration, or when an error occurred during frame transmission.

- Successful transmission:
  - Corresponding Tx Buffer Transmission Occurred bit TXBTO[TOx] set
  - Corresponding Tx Buffer Cancellation Finished bit TXBCF[CFx] not set
- Successful transmission in spite of cancellation:

- Corresponding Tx Buffer Transmission Occurred bit TXBTO[TOx] set
- Corresponding Tx Buffer Cancellation Finished bit TXBCF[CFx] set
- Arbitration lost or frame transmission disturbed:
  - Corresponding Tx Buffer Transmission Occurred bit TXBTO[TOx] not set
  - Corresponding Tx Buffer Cancellation Finished bit TXBCF[CFx] set

In case of a successful frame transmission, and if storage of Tx events is enabled, a Tx Event FIFO element is written with Event Type ET = 10 (transmission in spite of cancellation).

### 38.5.1.8 Power Down (Sleep Mode)

The M\_CAN can be set into power down mode controlled by CC Control Register CCCR[CSR].

When all pending transmission requests have completed, the M\_CAN waits until bus idle state is detected. Then the M\_CAN sets then CCCR[INIT] to one to prevent any further CAN transfers. Now the M\_CAN acknowledges that it is ready for power down by setting CCCR[CSA] to one. In this state, before the clocks are switched off, further register accesses can be made. A write access to CCCR[INIT] will have no effect. Now the module input clocks: CAN clock and Host clock may be switched off.

To leave power down mode, the application has to turn on the module clocks before resetting CC Control Register flag CCCR[CSR]. The M\_CAN will acknowledge this by resetting output signal clock stop acknowledge and resetting CCCR[CSA]. Afterwards, the application can restart CAN communication by resetting bit CCCR[INIT].

### 38.5.1.9 Test Modes

To enable write access to register TEST, bit CCCR[TEST] has to be set to one. This allows the configuration of the test modes and test functions.

Four output functions are available for the CAN transmit pin M\_CAN\_Tx by programming TEST[TX]. Additionally to its default function – the serial data output – it can drive the CAN Sample Point signal to monitor the M\_CAN's bit timing and it can drive constant dominant or recessive values. The actual value at pin M\_CAN\_Rx can be read from TEST[RX]. Both functions can be used to check the CAN bus' physical layer.

Due to the synchronization mechanism between CAN clock and Host clock domain, there may be a delay of several Host clock periods between writing to TEST[TX] until the new configuration is visible at output pin M\_CAN\_Tx. This applies also when reading input pin M\_CAN\_Rx via TEST[RX].

### Note

Test modes should be used for production tests or self test only.  
The software control for pin M\_CAN\_Tx interferes with all CAN protocol functions. It is not recommended to use test modes for application.

#### 38.5.1.9.1 External Loopback Mode

The M\_CAN can be set in External Loopback Mode by programming TEST[LBCK] to one. In Loopback Mode, the M\_CAN treats its own transmitted messages as received messages and stores them (if they pass acceptance filtering) into an Rx Buffer or an Rx FIFO. The following figure shows the connection of signals M\_CAN\_Tx and M\_CAN\_Rx to the M\_CAN in External Loopback Mode.

This mode is provided for hardware self-test. To be independent from external stimulation, the M\_CAN ignores acknowledge errors (recessive bit sampled in the acknowledge slot of a data/remote frame) in Loopback Mode. In this mode the M\_CAN performs an internal feedback from its Tx output to its Rx input. The actual value of the input pin M\_CAN\_Rx is disregarded by the M\_CAN. The transmitted messages can be monitored at the pin M\_CAN\_Tx.

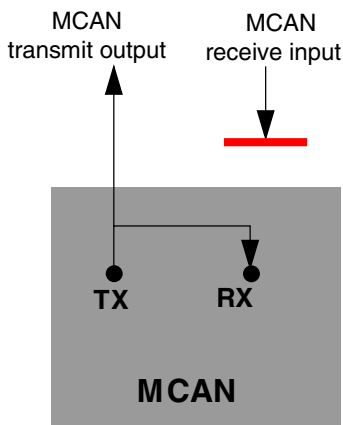


Figure 38-52. Pin Control in External Loopback Mode

### 38.5.1.9.2 Internal Loopback Mode

Internal Loopback Mode is entered by programming bits TEST[LBCK] and CCCR[MON] to one. This mode can be used for a "Hot Selftest", meaning the M\_CAN can be tested without affecting a running CAN system connected to the pins M\_CAN\_Tx and M\_CAN\_Rx. In this mode pin M\_CAN\_Rx is disconnected from the M\_CAN and pin M\_CAN\_Tx is held recessive. The following figure shows the connection of M\_CAN\_Tx and M\_CAN\_Rx to the M\_CAN in case of Internal Loopback Mode.

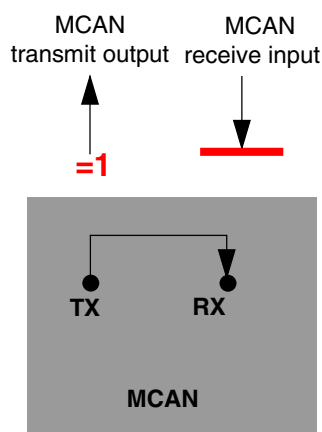


Figure 38-53. Pin Control in Internal Loopback Mode

### 38.5.2 Timestamp Generation

For timestamp generation the M\_CAN supplies a 16-bit wrap-around counter. A prescaler TSCC[TCP] can be configured to clock the counter in multiples of CAN bit times (1...16). The counter is readable via TSCV[TSC]. A write access to register TSCV resets the counter to zero. When the timestamp counter wraps around interrupt flag IR[TSW] is set.

On start of frame reception / transmission the counter value is captured and stored into the timestamp section of an Rx Buffer / Rx FIFO (RXTS[15:0]) or Tx Event FIFO (TXTS[15:0]) element.

### 38.5.3 Timeout Counter

To signal timeout conditions for Rx FIFO 0, Rx FIFO 1, and the Tx Event FIFO the M\_CAN supplies a 16-bit Timeout Counter. It operates as down-counter and uses the same prescaler controlled by TSCC[TCP] as the Timestamp Counter. The Timeout

Counter is configured via register TOCC. The actual counter value can be read from TOCV[TOC]. The Timeout Counter can only be started while CCCR[INIT] = 0. It is stopped when CCCR[INIT] = 1, e.g. when the M\_CAN enters Bus\_Off state.

The operation mode is selected by TOCC[TOS]. When operating in Continuous mode, the counter starts when CCCR[INIT] is reset. A write to TOCV presets the counter to the value configured by TOCC[TOP] and continues down-counting.

When the Timeout Counter is controlled by one of the FIFOs, an empty FIFO presets the counter to the value configured by TOCC[TOP]. Down-counting is started when the first FIFO element is stored. Writing to TOCV has no effect.

When the counter reaches zero, interrupt flag IR[TOO] is set. In Continuous Mode, the counter is immediately restarted at TOCC[TOP].

### **Note**

The clock signal for the Timeout Counter is derived from the CAN Core's sample point signal. Therefore the point in time where the Timeout Counter is decremented may vary due to the synchronization / re-synchronization mechanism of the CAN Core. If the baud rate switch feature in CAN FD is used, the timeout counter is clocked differently in arbitration and data field.

## **38.5.4 Rx Handling**

The Rx Handler controls the acceptance filtering, the transfer of received messages to the Rx Buffers or to one of the two Rx FIFOs, as well as the Rx FIFO's Put and Get Indices.

### **38.5.4.1 Acceptance filtering**

The M\_CAN offers the possibility to configure two sets of acceptance filters, one for standard identifiers and one for extended identifiers. These filters can be assigned to Rx Buffer or to Rx FIFO 0, 1. For acceptance filtering each list of filters is executed from element #0 until the first matching element. Acceptance filtering stops at the first matching element. The following filter elements are not evaluated for this message.

The main features are:

- Each filter element can be configured as
  - Range filter (from - to)



- Filter for one or two dedicated IDs
- Classic bit mask filter
- Each filter element is configurable for acceptance or rejection filtering
- Each filter element can be enabled / disabled individually
- Filters are checked sequentially, execution stops with the first matching filter element

Related configuration registers are:

- Global Filter Configuration (GFC)
- Standard ID Filter Configuration (SIDFC)
- Extended ID Filter Configuration (XIDFC)
- Extended ID AND Mask (XIDAM)

Depending on the configuration of the filter element (SFEC/EFEC) a match triggers one of the following actions:

- Store received frame in FIFO 0 or FIFO 1
- Store received frame in Rx Buffer
- Store received frame in Rx Buffer and generate pulse at filter event pin
- Reject received frame
- Set High Priority Message interrupt flag IR.HPM
- Set High Priority Message interrupt flag IR.HPM and store received frame in FIFO 0 or FIFO 1

Acceptance filtering is started after the complete identifier has been received. After acceptance filtering has completed, and if a matching Rx Buffer or Rx FIFO has been found, the Message Handler starts writing the received message data in portions of 32 bit to the matching Rx Buffer or Rx FIFO. If the CAN protocol controller has detected an error condition (e.g. CRC error), this message is discarded with the following impact on the affected Rx Buffer or Rx FIFO:

### **Rx Buffer**

New Data flag of matching Rx Buffer is not set, but Rx Buffer (partly) overwritten with received data. For error type see PSR[LEC] respectively PSR[DLEC].

### **Rx FIFO**

Put index of matching Rx FIFO is not updated, but related Rx FIFO element (partly) overwritten with received data. For error type see PSR[LEC] respectively PSR[DLEC]. In case the matching Rx FIFO is operated in overwrite mode, the boundary conditions described in "Rx FIFO Overwrite Mode" section have to be considered.

### **Note**

When an accepted message is written to one of the two Rx FIFOs, or into an Rx Buffer, the unmodified received identifier is stored independent of the filter(s) used. The result of the acceptance filter process is strongly depending on the sequence of configured filter elements.

#### **38.5.4.1.1 Range Filter**

The filter matches for all received frames with Message IDs in the range defined by SF1ID/SF2ID resp. EF1ID/EF2ID.

There are two possibilities when range filtering is used together with extended frames:

- EFT = "00": The Message ID of received frames is ANDed with the Extended ID AND Mask (XIDAM) before the range filter is applied
- EFT = "11": The Extended ID AND Mask (XIDAM) is not used for range filtering

#### **38.5.4.1.2 Filter for Specific IDs**

A filter element can be configured to filter for one or two specific Message IDs. To filter for one specific Message ID, the filter element has to be configured with SF1ID = SF2ID resp. EF1ID = EF2ID.

#### **38.5.4.1.3 Classic Bit Mask Filter**

Classic bit mask filtering is intended to filter groups of Message IDs by masking single bits of a received Message ID. With classic bit mask filtering SF1ID/EF1ID is used as Message ID filter, while SF2ID/EF2ID is used as filter mask.

A zero bit at the filter mask will mask out the corresponding bit position of the configured ID filter, e.g. the value of the received Message ID at that bit position is not relevant for acceptance filtering. Only those bits of the received Message ID where the corresponding mask bits are one are relevant for acceptance filtering.

In case all mask bits are one, a match occurs only when the received Message ID and the Message ID filter are identical. If all mask bits are zero, all Message IDs match.

#### 38.5.4.1.4 Standard Message ID Filtering

The following figure shows the flow for standard Message ID (11-bit Identifier) filtering. The Standard Message ID Filter element is described in [Standard Message ID Filter Element](#).

Controlled by the Global Filter Configuration GFC and the Standard ID Filter Configuration SIDFC Message ID, Remote Transmission Request bit (RTR), and the Identifier Extension bit (IDE) of received frames are compared against the list of configured filter elements.

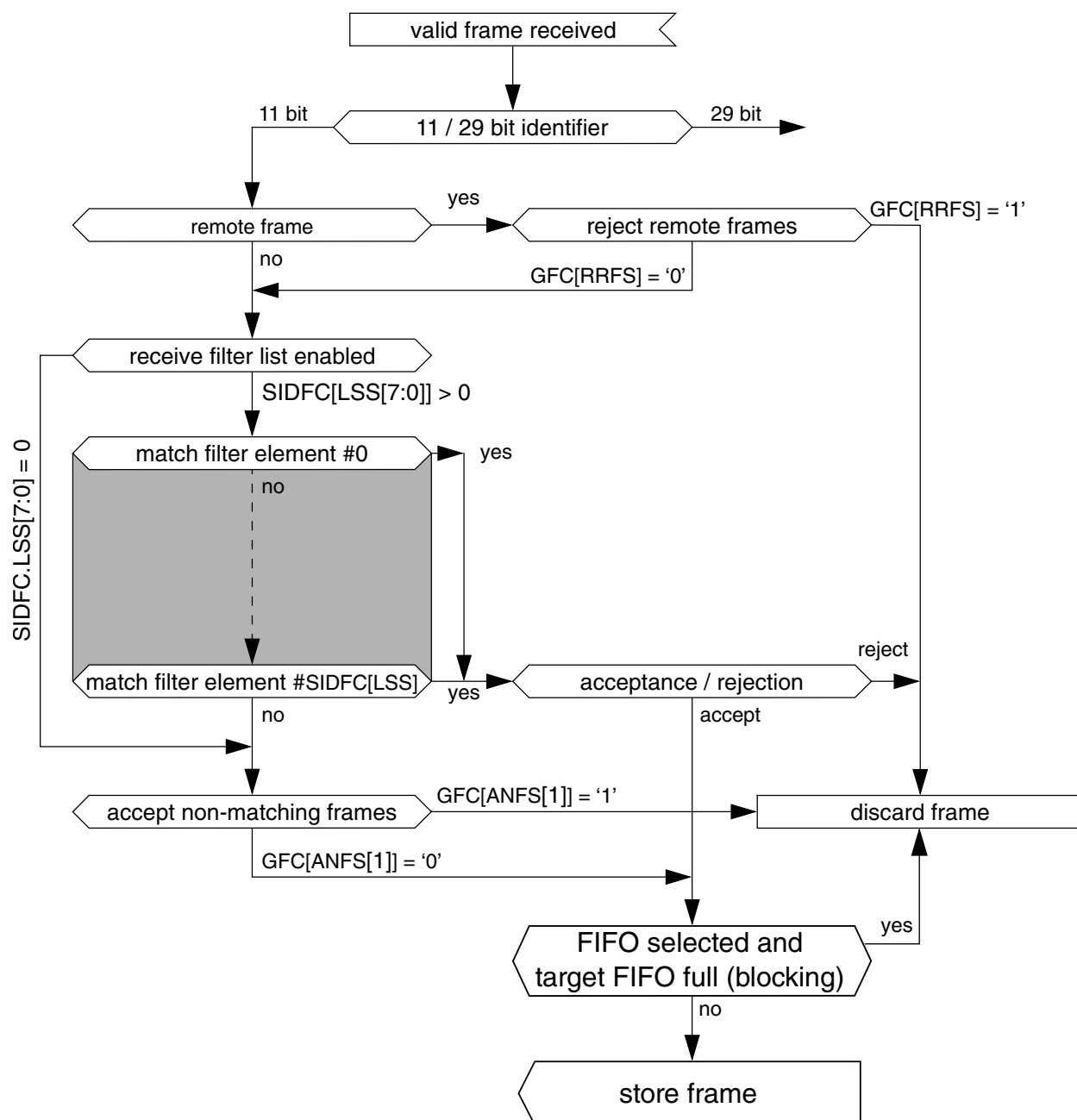


Figure 38-54. Standard Message ID Filter Path

### 38.5.4.1.5 Extended Message ID Filtering

The figure below shows the flow for extended Message ID (29-bit Identifier) filtering. The Extended Message ID Filter element is described in [Extended Message ID Filter Element](#).

Controlled by the Global Filter Configuration GFC and the Extended ID Filter Configuration XIDFC Message ID, Remote Transmission Request bit (RTR), and the Identifier Extension bit (IDE) of received frames are compared against the list of configured filter elements.

The Extended ID AND Mask XIDAM is ANDed with the received identifier before the filter list is executed.

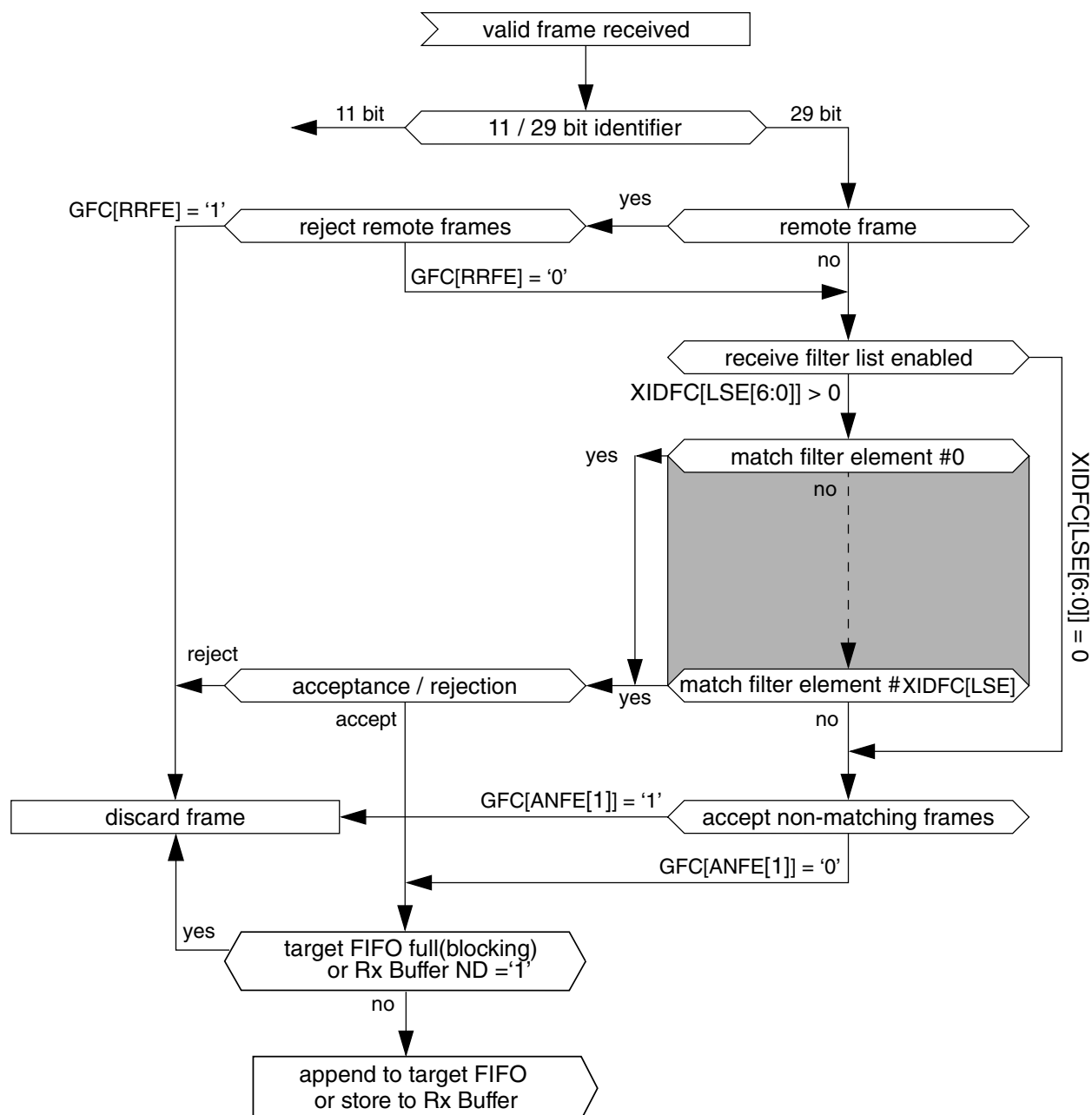


**Figure 38-55. Extended Message ID Filter Path**

### 38.5.4.1.6 Rx FIFOs

Rx FIFO 0 and Rx FIFO 1 can be configured to hold up to 64 elements each. Configuration of the two Rx FIFOs is done via registers RXF0C and RXF1C.

Received messages that passed acceptance filtering are transferred to the Rx FIFO as configured by the matching filter element. For a description of the filter mechanisms available for Rx FIFO 0 and Rx FIFO 1 see [Acceptance filtering](#). The Rx FIFO element is described in [Rx Buffer and FIFO Element](#).

To avoid an Rx FIFO overflow, the Rx FIFO watermark can be used. When the Rx FIFO fill level reaches the Rx FIFO watermark configured by RXFnC[FnWM], interrupt flag IR[RFnW] is set. When the Rx FIFO Put Index reaches the Rx FIFO Get Index an Rx FIFO Full condition is signalled by RXFnS[FnF]. In addition interrupt flag IR[RFnF] is set.

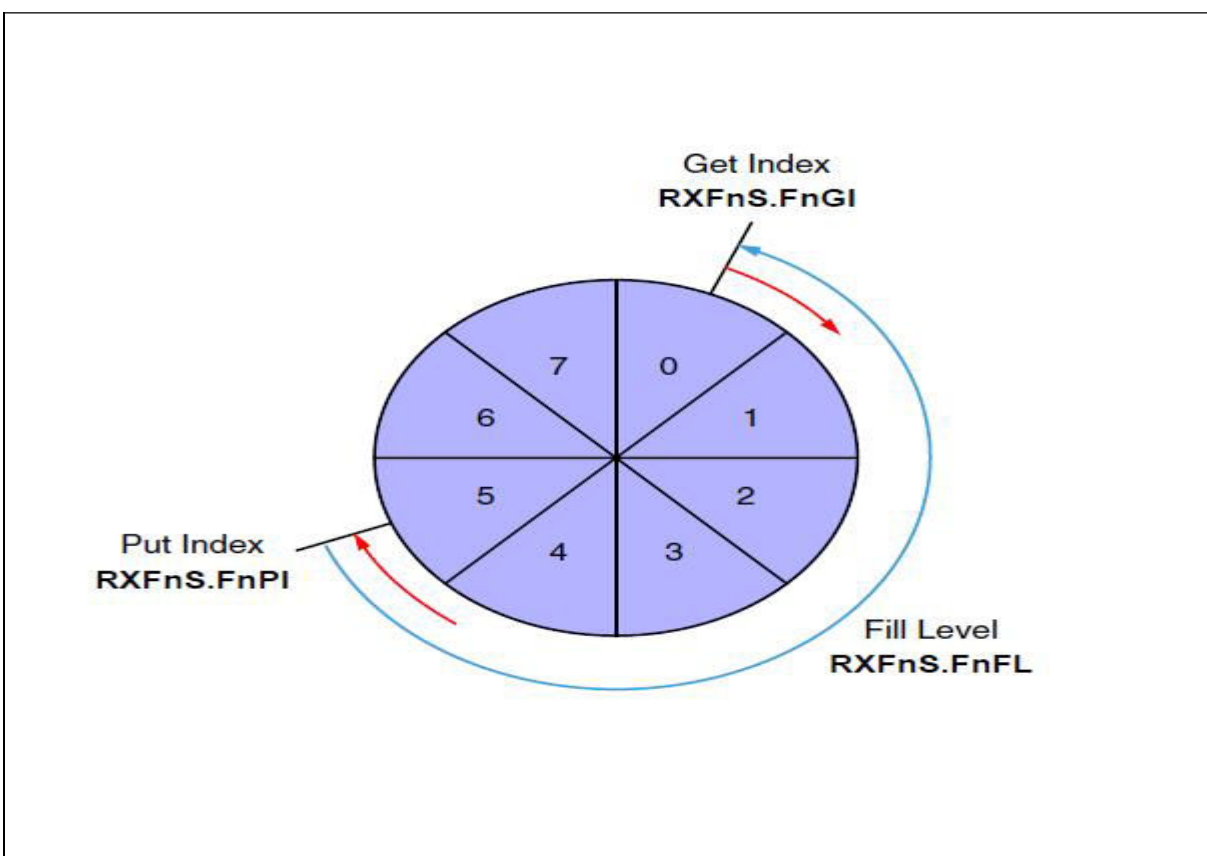


**Figure 38-56. Rx FIFO Status**

When reading from an Rx FIFO, Rx FIFO Get Index RXFnS[FnGI] x FIFO Element Size has to be added to the corresponding Rx FIFO start address RXFnC[FnSA].

**Table 38-60. Rx Buffer / FIFO Element Size**

| RXESC.RBDS[2:0] RXESC.FnDS[2:0] | Data Field [bytes] | FIFO Element Size [RAM words] |
|---------------------------------|--------------------|-------------------------------|
| 000                             | 8                  | 4                             |
| 001                             | 12                 | 5                             |
| 010                             | 16                 | 6                             |
| 011                             | 20                 | 7                             |
| 100                             | 24                 | 8                             |
| 101                             | 32                 | 10                            |
| 110                             | 48                 | 14                            |
| 111                             | 64                 | 18                            |

#### 38.5.4.1.6.1 Rx FIFO Blocking Mode

The Rx FIFO blocking mode is configured by  $RXFnC[FnOM] = 0$ . This is the default operation mode for the Rx FIFOs.

When an Rx FIFO full condition is reached ( $RXFnS[FnPI] = RXFnS[FnGI]$ ), no further messages are written to the corresponding Rx FIFO until at least one message has been read out and the Rx FIFO Get Index has been incremented. An Rx FIFO full condition is signalled by  $RXFnS[FnF] = 1$ . In addition interrupt flag  $IR[RFnF]$  is set.

In case a message is received while the corresponding Rx FIFO is full, this message is discarded and the message lost condition is signalled by  $RXFnS[RFnL] = 1$ . In addition interrupt flag  $IR[RFnL]$  is set.

#### 38.5.4.1.6.2 Rx FIFO Overwrite Mode

The Rx FIFO overwrite mode is configured by  $RXFnC[FnOM] = 1$ .

When an Rx FIFO full condition ( $RXFnS[FnPI] = RXFnS[FnGI]$ ) is signalled by  $RXFnS[FnF] = 1$ , the next message accepted for the FIFO will overwrite the oldest FIFO message. Put and get index are both incremented by one.

When an Rx FIFO is operated in overwrite mode and an Rx FIFO full condition is signalled, reading of the Rx FIFO elements should start at least at get index + 1. The reason for that is, that it might happen, that a received message is written to the Message RAM (put index) while the CPU is reading from the Message RAM (get index). In this case inconsistent data may be read from the respective Rx FIFO element. Adding an offset to the get index when reading from the Rx FIFO avoids this problem. The offset depends on how fast the CPU accesses the Rx FIFO. The following figure shows an offset of two with respect to the get index when reading the Rx FIFO. In this case the two messages stored in element 1 and 2 are lost.

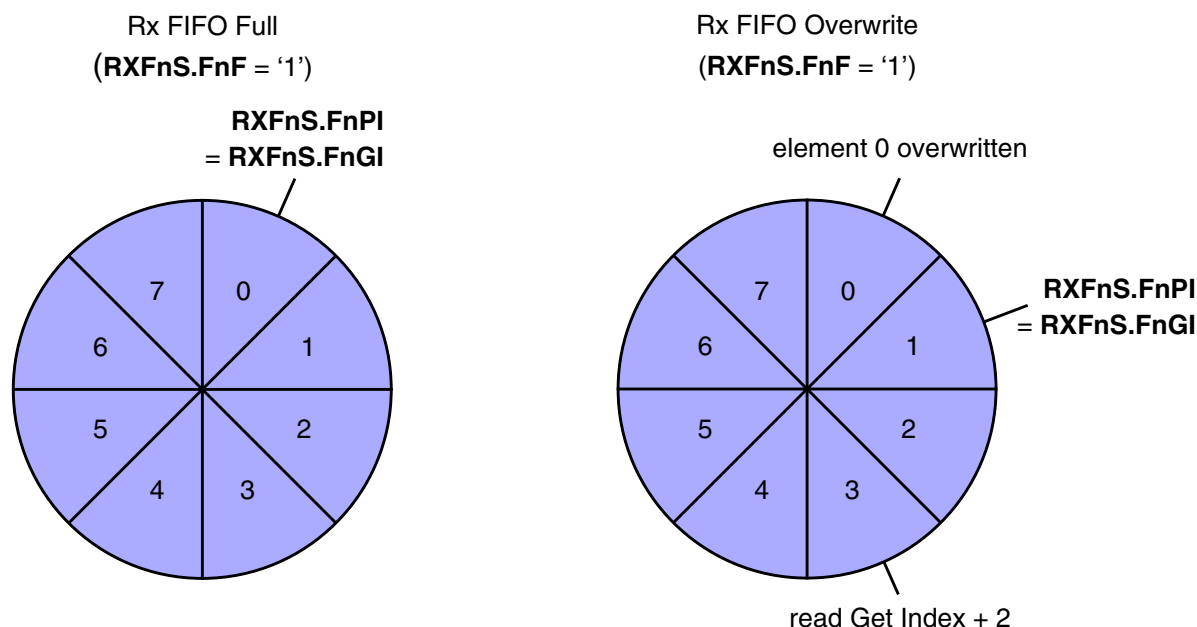


Figure 38-57. Rx FIFO Overflow Handling

After reading from the Rx FIFO, the number of the last element read has to be written to the Rx FIFO Acknowledge Index RXFnA[FnA]. This increments the get index to that element number. In case the put index has not been incremented to this Rx FIFO element, the Rx FIFO full condition is reset (RXFnS[FnF] = 0).

38.5.4.1.7 Dedicated Rx Buffers

The M\_CAN supports up to 64 dedicated Rx Buffers. The start address of the dedicated Rx Buffer section is configured via RXBC[RBSA].

For each Rx Buffer a Standard or Extended Message ID Filter Element with SFEC / EFEC = 111 and SFID2 / EFID2[10:9] = 00 has to be configured (see [Standard Message ID Filter Element](#) and [Extended Message ID Filter Element](#))

After a received message has been accepted by a filter element, the message is stored into the Rx Buffer in the Message RAM referenced by the filter element. The format is the same as for an Rx FIFO element. In addition the flag IR[DRX] (Message stored in Dedicated Rx Buffer) in the interrupt register is set.

Table 38-61. Example Filter Configuration for Rx buffers

| Filter Element | SFID1[10:0] EFID1[28:0] | SFID2[10:9] EFID2[10:9] | SFID2[5:0] EFID2[5:0] |
|----------------|-------------------------|-------------------------|-----------------------|
| 0              | ID message 1            | 00                      | 00 0000               |
| 1              | ID message 2            | 00                      | 00 0001               |
| 2              | ID message 3            | 00                      | 00 0010               |



After the last word of a matching received message has been written to the Message RAM, the respective New Data flag in register NDAT1,2 is set. As long as the New Data flag is set, the respective Rx Buffer is locked against updates from received matching frames. The New Data flags have to be reset by the Host by writing a 1 to the respective bit position.

While an Rx Buffer's New Data flag is set, a Message ID Filter Element referencing this specific Rx Buffer will not match, causing the acceptance filtering to continue. Following Message ID Filter Elements may cause the received message to be stored into another Rx Buffer, or into an Rx FIFO, or the message may be rejected, depending on filter configuration

#### **38.5.4.1.7.1 Rx Buffer Handling**

- Reset interrupt flag IR[DRX]
- Read New Data registers
- Read messages from Message RAM
- Reset New Data flags of processed messages

#### **38.5.4.1.8 Debug on CAN Support**

Debug messages are stored into Rx Buffers. For debug handling three consecutive Rx buffers (e.g. #61, #62, #63) have to be used for storage of debug messages A, B, and C. The format is the same as for an Rx Buffer or an Rx FIFO element.

Advantage: Fixed start address for the DMA transfers (relative to RXBC[RBSA]), no additional configuration required.

For filtering of debug messages Standard / Extended Filter Elements with SFEC / EFEC = 111 have to be set up. Messages matching these filter elements are stored into the Rx Buffers addressed by SFID2 / EFID2[5:0].

After message C has been stored, the DMA request output m\_can\_dma\_req is activated and the three messages can be read from the Message RAM under DMA control. The RAM words holding the debug messages will not be changed by the M\_CAN while m\_can\_dma\_req is activated. The behaviour is similar to that of an Rx Buffers with its New Data flag set.

After the DMA has completed the DMA unit sets m\_can\_dma\_ack. This resets m\_can\_dma\_req. Now the M\_CAN is prepared to receive the next set of debug messages.

### 38.5.4.1.8.1 Filtering for Debug Messages

Filtering for debug messages is done by configuring one Standard / Extended Message ID Filter Element for each of the three debug messages. To enable a filter element to filter for debug messages SFEC / EFEC has to be programmed to 111. In this case fields SFID1 / SFID2 and EFID1 / EFID2 have a different meaning (see [Standard Message ID Filter Element](#) and [Extended Message ID Filter Element](#)). While SFID2 / EFID2[10:9] controls the debug message handling state machine, SFID2 / EFID2[5:0] controls the location for storage of a received debug message.

When a debug message is stored, neither the respective New Data flag nor IR[DRX] are set. The reception of debug messages can be monitored via RXF1S[DMS].

**Table 38-62. Example Filter Configuration for Debug Messages**

| Filter Element | SFID1[10:0] EFID1[28:0] | SFID2[10:9] EFID2[10:9] | SFID2[5:0] EFID2[5:0] |
|----------------|-------------------------|-------------------------|-----------------------|
| 0              | ID debug message A      | 01                      | 11 1101               |
| 1              | ID debug message B      | 10                      | 11 1110               |
| 2              | ID debug message C      | 11                      | 11 1111               |

### 38.5.4.1.8.2 Debug Message Handling

The debug message handling state machine assures that debug messages are stored to three consecutive Rx Buffers in correct order. In case of missing messages the process is restarted. The DMA request is activated only when all three debug messages A, B, C have been received in correct order.

The status of the debug message handling state machine is signaled via RXF1S[DMS].

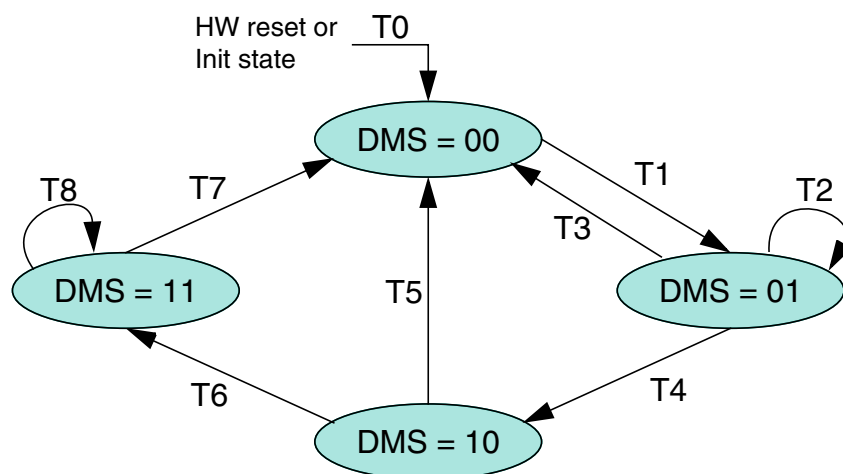


**Figure 38-58. Debug Message Handling State Machine**

T0: reset m\_can\_dma\_req output, enable reception of debug messages A, B, and C

T1: reception of debug message A

T2: reception of debug message A

T3: reception of debug message C

T4: reception of debug message B

T5: reception of debug messages A, B

T6: reception of debug message C

T7: DMA transfer completed

T8: reception of debug message A,B,C (message rejected)

### 38.5.5 Tx Handling

The Tx Handler handles transmission requests for the dedicated Tx Buffers, the Tx FIFO, and the Tx Queue. It controls the transfer of transmit messages to the CAN Core, the Put and Get Indices, and the Tx Event FIFO. Up to 32 Tx Buffers can be set up for message transmission. The CAN mode for transmission (Classic CAN or CAN FD) can be configured separately for each Tx Buffer element. The Tx Buffer element is described in [Tx Buffer Element](#). The table below describes the possible configurations for frame transmission.

**Table 38-63. Possible configurations for frame transmission**

| CCCR    |      | Tx Buffer Element |         | Frame Transmission            |
|---------|------|-------------------|---------|-------------------------------|
| BRSE    | FDOE | FDF               | BRS     |                               |
| ignored | 0    | ignored           | ignored | Classic CAN                   |
| 0       | 1    | 0                 | ignored | Classic CAN                   |
| 0       | 1    | 1                 | ignored | FD without bit rate switching |
| 1       | 1    | 0                 | ignored | Classic CAN                   |
| 1       | 1    | 1                 | 0       | FD without bit rate switching |
| 1       | 1    | 1                 | 1       | FD with bit rate switching    |

#### Note

AUTOSAR requires at least three Tx Queue Buffers and support of transmit cancellation.

The Tx Handler starts a Tx scan to check for the highest priority pending Tx request (Tx Buffer with lowest Message ID) when the Tx Buffer Request Pending register TXBRP is updated, or when a transmission has been started.

### 38.5.5.1 Transmit Pause

The transmit pause feature is intended for use in CAN systems where the CAN message identifiers are (permanently) specified to specific values and cannot easily be changed. These message identifiers may have a higher CAN arbitration priority than other defined messages, while in a specific application their relative arbitration priority should be inverse. This may lead to a case where one ECU sends a burst of CAN messages that cause another ECU's CAN messages to be delayed because that other messages have a lower CAN arbitration priority.

If, for example, CAN ECU-1 has the transmit pause feature enabled and is requested by its application software to transmit four messages, it will, after the first successful message transmission, wait for two CAN bit times of bus idle before it is allowed to start the next requested message. If there are other ECUs with pending messages, those messages are started in the idle time, they would not need to arbitrate with the next message of ECU-1. After having received a message, ECU-1 is allowed to start its next transmission as soon as the received message releases the CAN bus.

The transmit pause feature is controlled by bit CCCR.TXP. If the bit is set, the M\_CAN will, each time it has successfully transmitted a message, pause for two CAN bit times before starting the next transmission. This enables other CAN nodes in the network to transmit messages even if their messages have lower prior identifiers. Default is transmit pause disabled (CCCR[TXP] = 0).

This feature looses up burst transmissions coming from a single node and it protects against “babbling idiot” scenarios where the application program erroneously requests too many transmissions.

### 38.5.5.2 Dedicated Tx Buffers

Dedicated Tx Buffers are intended for message transmission under complete control of the Host CPU. Each Dedicated Tx Buffer is configured with a specific Message ID. In case that multiple Tx Buffers are configured with the same Message ID, the Tx Buffer with the lowest buffer number is transmitted first.

If the data section has been updated, a transmission is requested by an Add Request via TXBAR[ARn]. The requested messages arbitrate internally with messages from an optional Tx FIFO or Tx Queue and externally with messages on the CAN bus, and are sent out according to their Message ID.

A Dedicated Tx Buffer allocates Element Size 32-bit words in the Message RAM (see the following table). Therefore the start address of a Dedicated Tx Buffer in the Message RAM is calculated by adding transmit buffer index (0...31) x Element Size to the Tx Buffer Start Address TXBC[TBSA].

**Table 38-64. Tx Buffer / FIFO / Queue Element Size**

| TXESC.TBDS[2:0] | Data Field [bytes] | Element Size [RAM words] |
|-----------------|--------------------|--------------------------|
| 000             | 8                  | 4                        |
| 001             | 12                 | 5                        |
| 010             | 16                 | 6                        |
| 011             | 20                 | 7                        |
| 100             | 24                 | 8                        |
| 101             | 32                 | 10                       |
| 110             | 48                 | 14                       |
| 111             | 64                 | 18                       |

### 38.5.5.3 Tx FIFO

Tx FIFO operation is configured by programming TXBC[TFQM] to 0. Messages stored in the Tx FIFO are transmitted starting with the message referenced by the Get Index TXFQS[TFGI]. After each transmission the Get Index is incremented cyclically until the Tx FIFO is empty. The Tx FIFO enables transmission of messages with the same Message ID from different Tx Buffers in the order these messages have been written to the Tx FIFO. The M\_CAN calculates the Tx FIFO Free Level TXFQS[TFFL] as difference between Get and Put Index. It indicates the number of available (free) Tx FIFO elements.

New transmit messages have to be written to the Tx FIFO starting with the Tx Buffer referenced by the Put Index TXFQS[TFQPI]. An Add Request increments the Put Index to the next free Tx FIFO element. When the Put Index reaches the Get Index, Tx FIFO Full (TXFQS[TFQF] = 1) is signalled. In this case no further messages should be written to the Tx FIFO until the next message has been transmitted and the Get Index has been incremented.

When a single message is added to the Tx FIFO, the transmission is requested by writing a 1 to the TXBAR bit related to the Tx Buffer referenced by the Tx FIFO's Put Index.

When multiple (n) messages are added to the Tx FIFO, they are written to n consecutive Tx Buffers starting with the Put Index. The transmissions are then requested via TXBAR. The Put Index is then cyclically incremented by n. The number of requested Tx buffers should not exceed the number of free Tx Buffers as indicated by the Tx FIFO Free Level.

When a transmission request for the Tx Buffer referenced by the Get Index is cancelled, the Get Index is incremented to the next Tx Buffer with pending transmission request and the Tx FIFO Free Level is recalculated. When transmission cancellation is applied to any other Tx Buffer, the Get Index and the FIFO Free Level remain unchanged.

A Tx FIFO element allocates Element Size 32-bit words in the Message RAM (See [Table 38-64](#)). Therefore the start address of the next available (free) Tx FIFO Buffer is calculated by adding Tx FIFO/Queue Put Index TXFQS.TFQPI (0...31) x Element Size to the Tx Buffer Start Address TXBC[TBSA].

#### 38.5.5.4 Tx Queue

Tx Queue operation is configured by programming TXBC[TFQM] to 1. Messages stored in the Tx Queue are transmitted starting with the message with the lowest Message ID (highest priority). In case that multiple Queue Buffers are configured with the same Message ID, the Queue Buffer with the lowest buffer number is transmitted first.

New messages have to be written to the Tx Buffer referenced by the Put Index TXFQS[TFQPI]. An Add Request cyclically increments the Put Index to the next free Tx Buffer. In case that the Tx Queue is full (TXFQS[TFQF] = 1), the Put Index is not valid and no further message should be written to the Tx Queue until at least one of the requested messages has been sent out or a pending transmission request has been cancelled.

The application may use register TXBRP instead of the Put Index and may place messages to any Tx Buffer without pending transmission request.

A Tx Queue Buffer allocates Element Size 32-bit words in the Message RAM (see [Table 38-64](#)). Therefore the start address of the next available (free) Tx Queue Buffer is calculated by adding Tx FIFO/Queue Put Index TXFQS[TFQPI] (0...31) x Element Size to the Tx Buffer Start Address TXBC[TBSA].

### 38.5.5.5 Mixed Dedicated Tx Buffers / Tx FIFO

In this case the Tx Buffers section in the Message RAM is subdivided into a set of Dedicated Tx Buffers and a Tx FIFO. The number of Dedicated Tx Buffers is configured by TXBC[NDTB]. The number of Tx Buffers assigned to the Tx FIFO is configured by TXBC[TFQS]. In case TXBC[TFQS] is programmed to zero, only Dedicated Tx Buffers are used.

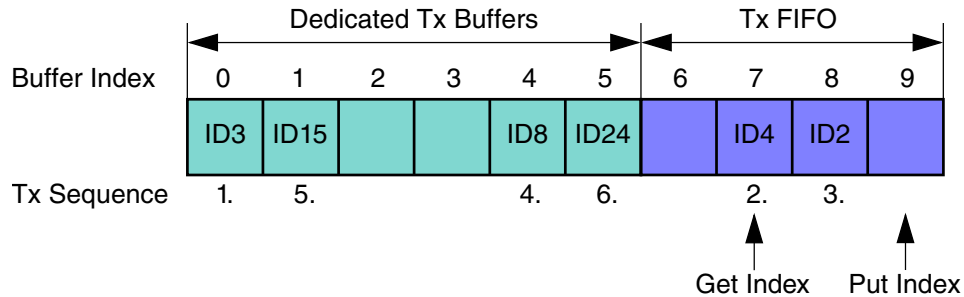


**Figure 38-59. Example of mixed Configuration Dedicated Tx Buffers / Tx FIFO**

Tx prioritization:

- Scan Dedicated Tx Buffers and oldest pending Tx FIFO Buffer (referenced by TXFS[TFGI])
- Buffer with lowest Message ID gets highest priority and is transmitted next

### 38.5.5.6 Mixed Dedicated Tx Buffers / Tx Queue

In this case the Tx Buffers section in the Message RAM is subdivided into a set of Dedicated Tx Buffers and a Tx Queue. The number of Dedicated Tx Buffers is configured by TXBC.NDTB. The number of Tx Queue Buffers is configured by TXBC.TFQS. In case TXBC.TFQS is programmed to zero, only Dedicated Tx Buffers are used.

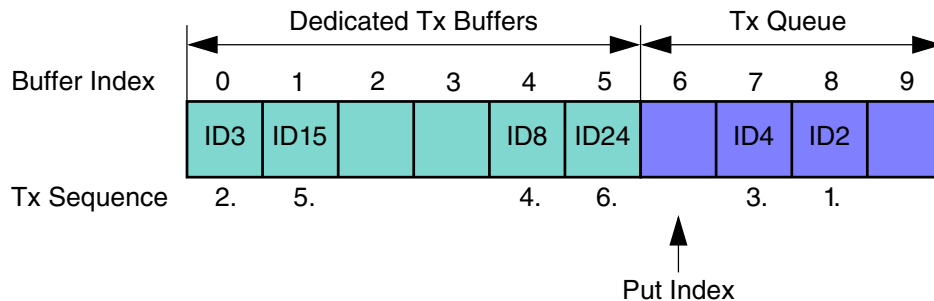


**Figure 38-60. Example of mixed Configuration Dedicated Tx Buffers / Tx Queue**

Tx prioritization:

- Scan all Tx Buffers with activated transmission request
- Tx Buffer with lowest Message ID gets highest priority and is transmitted next

### 38.5.5.7 Transmit Cancellation

The M\_CAN supports transmit cancellation. This feature is especially intended for gateway applications and AUTOSAR based applications. To cancel a requested transmission from a dedicated Tx Buffer or a Tx Queue Buffer the Host has to write a 1 to the corresponding bit position (= number of Tx Buffer) of register TXBCR. Transmit cancellation is not intended for Tx FIFO operation.

Successful cancellation is signalled by setting the corresponding bit of register TXBCF to 1.

In case a transmit cancellation is requested while a transmission from a Tx Buffer is already ongoing, the corresponding TXBRP bit remains set as long as the transmission is in progress. If the transmission was successful, the corresponding TXBTO and TXBCF bits are set. If the transmission was not successful, it is not repeated and only the corresponding TXBCF bit is set.

#### Note

In case a pending transmission is cancelled immediately before this transmission could have been started, there follows a short time window where no transmission is started even if another message is also pending in this node. This may enable another node to transmit a message which may have a lower priority than the second message in this node.

### 38.5.5.8 Tx Event Handling

To support Tx event handling the M\_CAN has implemented a Tx Event FIFO. After the M\_CAN has transmitted a message on the CAN bus, Message ID and timestamp are stored in a Tx Event FIFO element. To link a Tx event to a Tx Event FIFO element, the Message Marker from the transmitted Tx Buffer is copied into the Tx Event FIFO element.

The Tx Event FIFO can be configured to a maximum of 32 elements. The Tx Event FIFO element is described in [Tx Event FIFO Element](#).



The purpose of the Tx Event FIFO is to decouple handling transmit status information from transmit message handling i.e. a Tx Buffer holds only the message to be transmitted, while the transmit status is stored separately in the Tx Event FIFO. This has the advantage, especially when operating a dynamically managed transmit queue, that a Tx Buffer can be used for a new message immediately after successful transmission. There is no need to save transmit status information from a Tx Buffer before overwriting that Tx Buffer.

When a Tx Event FIFO full condition is signalled by IR[TEFF], no further elements are written to the Tx Event FIFO until at least one element has been read out and the Tx Event FIFO Get Index has been incremented. In case a Tx event occurs while the Tx Event FIFO is full, this event is discarded and interrupt flag IR[TEFL] is set.

To avoid a Tx Event FIFO overflow, the Tx Event FIFO watermark can be used. When the Tx Event FIFO fill level reaches the Tx Event FIFO watermark configured by TXEFC[EFWM], interrupt flag IR[TEFW] is set.

When reading from the Tx Event FIFO, two times the Tx Event FIFO Get Index TXEFS[EFGI] has to be added to the Tx Event FIFO start address TXEFC[EFSA].

### 38.5.6 FIFO Acknowledge Handling

The Get Indices of Rx FIFO 0, Rx FIFO 1, and the Tx Event FIFO are controlled by writing to the corresponding FIFO Acknowledge Index (see the registers RXF0A, RXF1A, and TXEFA). Writing to the FIFO Acknowledge Index will set the FIFO Get Index to the FIFO Acknowledge Index plus one and thereby updates the FIFO Fill Level. There are two use cases:

- When only a single element has been read from the FIFO (the one being pointed to by the Get Index), this Get Index value is written to the FIFO Acknowledge Index.
- When a sequence of elements has been read from the FIFO, it is sufficient to write the FIFO Acknowledge Index only once at the end of that read sequence (value: Index of the last element read), to update the FIFO's Get Index.

Due to the fact that the CPU has free access to the M\_CAN's Message RAM, special care has to be taken when reading FIFO elements in an arbitrary order (Get Index not considered). This might be useful when reading a High Priority Message from one of the two Rx FIFOs. In this case the FIFO's Acknowledge Index should not be written because this would set the Get Index to a wrong position and also alters the FIFO's Fill Level. In this case some of the older FIFO elements would be lost.

### Note

The application has to ensure that a valid value is written to the FIFO Acknowledge Index. The M\_CAN does not check for erroneous values.

# Chapter 39

## Deserial Serial Peripheral Interface (DSPI)

### 39.1 Chip-specific DSPI information

This chip implements five instances of the DSPI module. Some feature details vary among the instances.

The following table summarizes the feature differences. The table does not list feature details that all instances share.

**Table 39-1. DSPI instance feature differences**

| Feature                                                                                                    | DSPI_A | DSPI_B | DSPI_C | DSPI_D     | DSPI_E          |
|------------------------------------------------------------------------------------------------------------|--------|--------|--------|------------|-----------------|
| Serialization mechanisms defined in the SIU chapter                                                        | Yes    | Yes    | Yes    | Yes        | No <sup>1</sup> |
| Serial Data In (SIN) signals                                                                               | Yes    | Yes    | Yes    | Yes        | No              |
| PISRs and bit muxes are implemented                                                                        | Yes    | Yes    | Yes    | Yes        | No <sup>2</sup> |
| Hardware Trigger (HT)                                                                                      | Yes    | Yes    | Yes    | Yes        | No              |
| Clock and data signals (SCK and SOUT) are available on LVDS pairs to support microsecond bus functionality | No     | No     | Yes    | Yes        | No              |
| Output serialization bits used                                                                             | None   | [15:0] | [15:0] | [15:4,1:0] | None            |

1. DSPI\_E does not support serialization, except what is provided by the ASDR register.

2. The registers PSIR0, PSIR1, PSIR2, and PISR3 do not exist for DSPI\_E.

#### 39.1.1 DSPI and eMIOS direct connections

The following table shows direct connections between eMIOS and DSPI module instances.

**Table 39-2. DSPI and eMIOS direct connections**

| Source | Destination | Description                                                                |
|--------|-------------|----------------------------------------------------------------------------|
| DSPI_B | eMIOS_0     | Connection from DSPI_B deserial data out bit 14 to eMIOS_0 input channel 5 |
| DSPI_B | eMIOS_0     | Connection from DSPI_B deserial data out bit 15 to eMIOS_0 input channel 4 |
| DSPI_D | eMIOS_1     | Connection from DSPI_D deserial data out bit 14 to eMIOS_1 input channel 7 |
| DSPI_D | eMIOS_1     | Connection from DSPI_D deserial data out bit 15 to eMIOS_1 input channel 6 |

### 39.1.2 DSPI connections with SIU, eTPU, and eMIOS modules

Some DSPI deserialized outputs can be used to trigger external interrupt requests via the External Interrupt Input Multiplexing sub-block in the SIU. See [External IRQ Input Select Register \(SIU\\_EIISR\)](#) for details.

DSPI\_A, DSPI\_B, DSPI\_C, and DSPI\_D connect via SIU registers to channels of eTPU and eMIOS modules as well as GPO data registers. See [Pin serialization for DSPI](#) for more information. The following table lists the applicable SIU registers for each DSPI instance.

**Table 39-3. SIU registers for DSPI pin serialization**

| DSPI instance | SIU registers                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DSPI_A        | <a href="#">Mask-Output High Register (SIU_DSPIAH)</a><br><a href="#">Mask-Output Low Register (SIU_DSPIAL)</a><br><a href="#">Serialized Output Signal Selection Register for DSPI_A (SIU_ETPUBA)</a><br><a href="#">Serialized Output Signal Selection Register for DSPI_A (SIU_EMIOSA)</a><br><a href="#">Serialized Output Signal Selection Register for DSPI_A (SIU_DSPIAHLA)</a>                                                                                        |
| DSPI_B        | <a href="#">Mask-Output High Register (SIU_DSPIBH)</a><br><a href="#">Mask-Output Low Register (SIU_DSPIBL)</a><br><a href="#">Serialized Output Signal Selection Register for DSPI_B (SIU_ETPUAB)</a><br><a href="#">Serialized Output Signal Selection Register for DSPI_B (SIU_EMIOSB)</a><br><a href="#">Serialized Output Signal Selection Register for DSPI_B (SIU_DSPIBHLB)</a>                                                                                        |
| DSPI_C        | <a href="#">Mask-Output High Register (SIU_DSPICH)</a><br><a href="#">Mask-Output Low Register (SIU_DSPICL)</a><br><a href="#">Serialized Output Signal Selection Register for DSPI_C (SIU_ETPUAC)</a><br><a href="#">Serialized Output Signal Selection Register for DSPI_C (SIU_EMIOSC)</a><br><a href="#">Serialized Output Signal Selection Register for DSPI_C (SIU_DSPICHLA)</a><br><a href="#">Serialized Output Signal Selection Register for DSPI_C (SIU_ETPUBC)</a> |
| DSPI_D        | <a href="#">Mask-Output High Register (SIU_DSPIDH)</a><br><a href="#">Mask-Output Low Register (SIU_DSPIDL)</a><br><a href="#">Serialized Output Signal Selection Register for DSPI_D (SIU_ETPUBD)</a>                                                                                                                                                                                                                                                                        |

**Table 39-3. SIU registers for DSPI pin serialization**

| DSPI instance | SIU registers                                                         |
|---------------|-----------------------------------------------------------------------|
|               | Serialized Output Signal Selection Register for DSPI_D (SIU_EMIOSD)   |
|               | Serialized Output Signal Selection Register for DSPI_D (SIU_DSPIDHLD) |

## 39.2 Introduction

The deserial serial peripheral interface (DSPI) module provides a synchronous serial bus for communication between a chip and an external peripheral device. The module supports chip pin count reduction through serialization and deserialization of chip internal signals transmitted over the D SPI serial link.

### 39.2.1 Block Diagram

The block diagram of this module is as follows:

#### NOTE

SPI is used for both SPI Mode and DSI mode in this diagram.

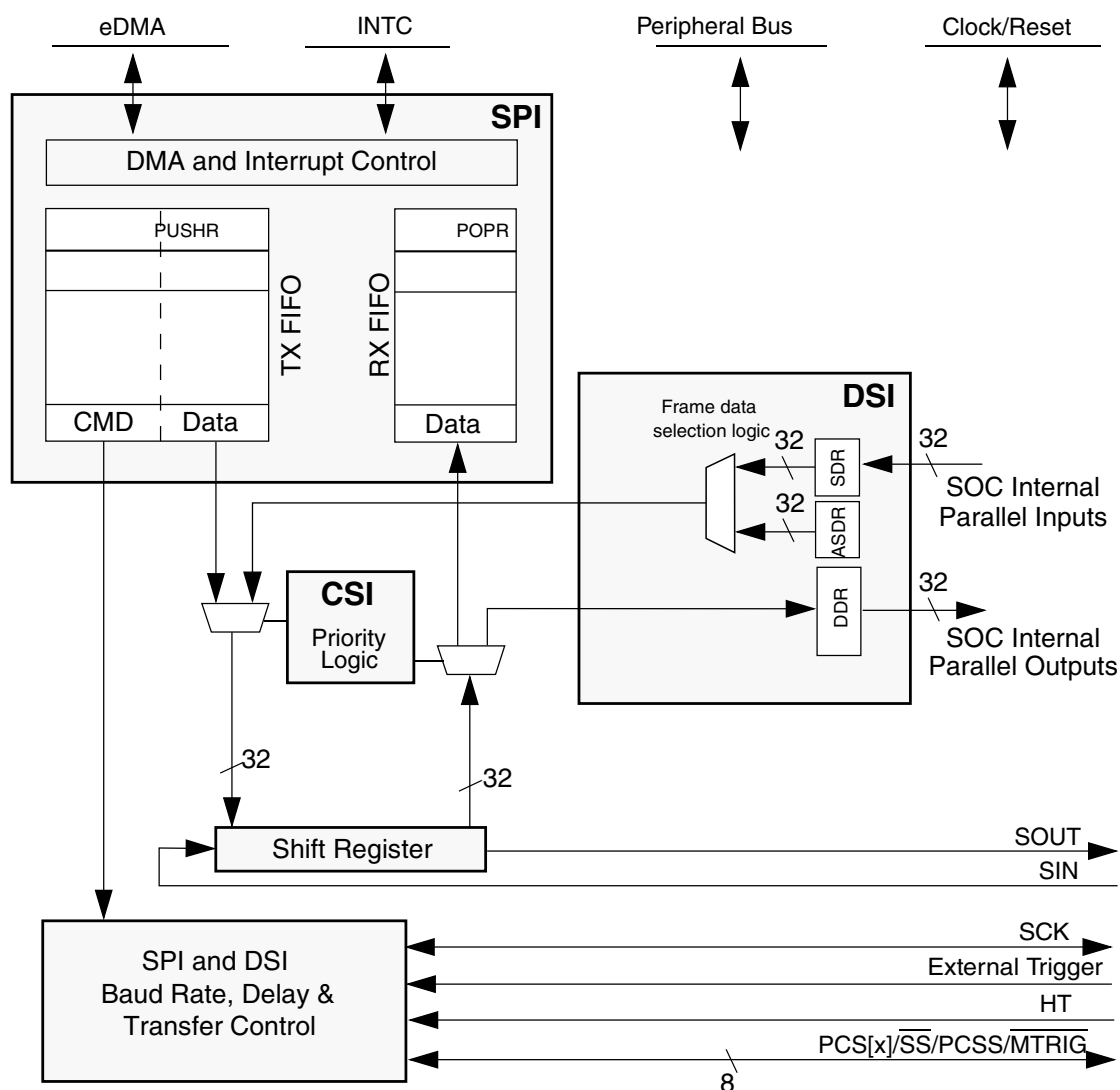


Figure 39-1. DSPI Block Diagram

### 39.2.2 Features

The module supports the following features:

- Full-duplex, three-wire synchronous transfers
- Master mode
- Slave mode
- Data streaming operation in Slave mode with continuous slave selection
- Buffered transmit operation using the transmit first in first out (TX FIFO) with depth of 4 entries

- Buffered receive operation using the receive FIFO (RX FIFO) with depth of 4 entries
- Asynchronous clocking scheme for Register and Protocol Interfaces
- TX and RX FIFOs can be disabled individually for low-latency updates to DSPI queues
- Visibility into TX and RX FIFOs for ease of debugging
- Programmable transfer attributes on a per-frame basis:
  - eight transfer attribute registers
  - Serial clock (SCK) with programmable polarity and phase
  - Various programmable delays
  - Programmable serial frame size: 4 to 32
    - DSPI frames longer than 16 bits can be supported using the continuous selection format.
  - Continuously held chip select capability
  - Parity control
- 6 peripheral chip selects (PCSeS), expandable to 64 with external demultiplexer
- Deglitching support for up to 32 peripheral chip selects (PCSeS) with external demultiplexer
- DMA support for adding entries to TX FIFO and removing entries from RX FIFO:
  - TX FIFO is not full (TFFF)
  - RX FIFO is not empty (RFDF)
- Interrupt conditions:
  - End of Queue reached (EOQF)
  - TX FIFO is not full (TFFF)
  - Transfer of current frame complete (TCF)
  - Attempt to transmit with an empty Transmit FIFO (TFUF)
  - RX FIFO is not empty (RFDF)
  - Frame received while Receive FIFO is full (RFOF)
  - DSPI Parity Error (SPEF)

- Modified DSPI transfer formats for communication with slower peripheral devices
- Power-saving architectural features:
  - Support for Stop mode

The DSPI also supports pin reduction through serialization and deserialization if enabled for the module.

- Two sources of serialized data:
  - The module memory-mapped register
  - Parallel Input signals
  - Programmable selection of source data on bit basis
- Deserialized data is provided as Parallel Output signals and as bits in a memory-mapped register
- Interrupt conditions:
  - Deserialized data matches pre-programmed pattern (DDIF)
  - DSI Parity Error (DPEF)
- Transfer initiation conditions:
  - Continuous
  - Edge sensitive hardware trigger
  - Change in data
- Pin serialization/deserialization with interleaved DSPI frames for control and diagnostics
- Continuous serial communications clock

DSPI supports a combined DSPI and DSI modes of operation known as the Combined Serial Interface (CSI). It also supports the downstream Micro Second Channel in the Timed Serial Bus (TSB) configuration.

- Transmission of frames is done at frame boundaries
- Frames from DSPI and DSI are identifiable by a bit transmitted at the start of each frame
- Separate interrupts for frame completion from DSPI and DSI



### 39.2.3 Interface configurations

#### 39.2.3.1 DSPI configuration

The Serial Peripheral Interface DSPI configuration allows the module to send and receive serial data. This configuration allows the module to operate as a basic DSPI block with internal FIFOs supporting external queue operation. Transmitted data and received data reside in separate FIFOs. The host CPU or a DMA controller read the received data from the Receive FIFO and write transmit data to the Transmit FIFO.

For queued operations, the SPI queues can reside in system RAM, external to the module. Data transfers between the queues and the module FIFOs are accomplished by a DMA controller or host CPU. The following figure shows a system example with DMA, DSPI, and external queues in system RAM.

#### NOTE

SPI is used for both DSPI Mode and DSI mode in this diagram.

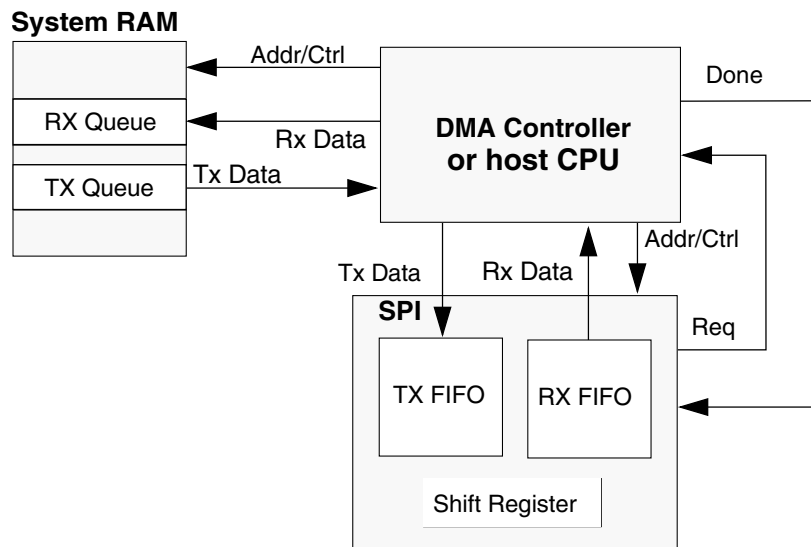


Figure 39-2. SPI with queues and DMA

#### 39.2.3.2 DSI configuration

In the Deserial Serial Interface (DSI) configuration, DSPI serializes up to 32 parallel input signals or register bits. DSPI also deserializes the received data to parallel output signals or to a memory-mapped register. DSPI transfers the data using an DSPI-like protocol.

Timed Serial Bus (TSB) mode provides the MicroSecond Channel (MSC) downstream support, serializing from 4 to 32 parallel input signals or register bits. For more information on TSB mode, see [Timed Serial Bus \(TSB\)](#).

### **39.2.3.3 CSI configuration**

The Combined Serial Interface (CSI) configuration supports further configuration as follows:

- Normal: A combination of the DSPI and DSI configurations. The DSPI interleaves DSI data frames with data frames. Interleaving is done on the frame boundaries.
- TSB: Transmission of DSPI data has higher priority than DSI data.

## **39.2.4 Modes of Operation**

The module supports the following modes of operation that can be divided into two categories:

- Module-specific modes:
  - Master mode
  - Slave mode
  - Module Disable mode
- Chip-specific modes:
  - External Stop mode

The module enters module-specific modes when the host writes a module register. The chip-specific modes are controlled by signals external to the module. The chip-specific modes are modes that a chip may enter in parallel to the block-specific modes.

### **39.2.4.1 Master Mode**

Master mode allows the module to initiate and control serial communication. In this mode, these signals are controlled by the module and configured as outputs:

- SCK
- PCS[x]

### 39.2.4.2 Slave Mode

Slave mode allows the module to communicate with DSPI/DSI bus masters. In this mode, the module responds to externally controlled serial transfers. The SCK signal and the PCS[0]/ $\overline{\text{SS}}$  signals are configured as inputs and driven by an SPI bus master.

### 39.2.4.3 Module Disable Mode

The Module Disable mode can be used for chip power management. The clock to the non-memory mapped logic in the module can be stopped while in the Module Disable mode.

### 39.2.4.4 External Stop Mode

External Stop mode is used for chip power management. The module supports the Peripheral Bus Stop mode mechanism. When a request is made to enter External Stop mode, it acknowledges the request and completes the transfer that is in progress. When the module reaches the frame boundary, it signals that the protocol clock to the module may be shut off.

#### NOTE

In master mode, if a serial transfer is in progress, then this module waits until it reaches the frame boundary before it is ready to have its clocks shut off. In slave mode, this module waits until its chip select input signal goes high before it is ready to have its clocks shut off. It should be noted that the CS0 pad must be correctly configured to allow it to return to high when no transfer is occurring. If the pad is tied low then this could prevent the module from being correctly disabled.

## 39.3 Module signal descriptions

This table describes the signals on the boundary of the module that may connect off chip (in alphabetical order).

**Table 39-4. Module signal descriptions**

| Signal  | Master mode                  | Slave mode       | I/O |
|---------|------------------------------|------------------|-----|
| HT      | Hardware Trigger             | Hardware Trigger | I   |
| PCS0/SS | Peripheral Chip Select 0 (O) | Slave Select (I) | I/O |

*Table continues on the next page...*

**Table 39-4. Module signal descriptions (continued)**

| Signal                         | Master mode                                             | Slave mode       | I/O |
|--------------------------------|---------------------------------------------------------|------------------|-----|
| PCS[1:3]                       | Peripheral Chip Selects 1–3                             | (Unused)         | O   |
| PCS4                           | Peripheral Chip Select 4                                | (Unused)         | O   |
| PCS5/ $\overline{\text{PCSS}}$ | Peripheral Chip Select 5 /Peripheral Chip Select Strobe | (Unused)         | O   |
| SCK                            | Serial Clock (O)                                        | Serial Clock (I) | I/O |
| SIN                            | Serial Data In                                          | Serial Data In   | I   |
| SOUT                           | Serial Data Out                                         | Serial Data Out  | O   |

### 39.3.1 HT—Hardware Trigger

Master mode: Hardware Trigger (I)—Honored only when the module is configured for the DSI or CSI configuration (MCR[DCONF]), hardware-trigger reception is enabled (DSICR0[TRRE] or TRIG[TRRE]), and the module is in the Running state (SR[TXRXS]). Receives hardware triggers that initiate data transfers. DSICR0[TPOL] specifies whether the module uses rising or falling edges of HT as hardware triggers.

Slave mode: Hardware Trigger (I)—Honored only when the module is configured for the DSI or CSI configuration (MCR[DCONF]), hardware-trigger reception is enabled (DSICR0[TRRE] or TRIG[TRRE]), and the module is in the Running state (SR[TXRXS]). DSICR0[TPOL] specifies whether the module uses rising or falling edges of HT as hardware triggers.

### 39.3.2 PCS0/ $\overline{\text{SS}}$ —Peripheral Chip Select/Slave Select

Master mode: Peripheral Chip Select 0 (O)—Selects an DSPI slave to receive data transmitted from the module.

Slave mode: Slave Select (I)—Selects the module to receive data transmitted from an DSPI master.

#### NOTE

Do not tie the DSPI slave select pin to ground. Otherwise, DSPI cannot function properly.

### 39.3.3 PCS1–PCS3—Peripheral Chip Selects 1–3

Master mode: Peripheral Chip Selects 1–3 (O)—Select an DSPI slave to receive data transmitted by the module.

Slave mode: Unused

### 39.3.4 PCS4—Peripheral Chip Select 4

Master mode: Peripheral Chip Select 4 (O)—Selects an DSPI slave to receive data transmitted by the module.

Slave mode: Unused

### 39.3.5 PCS5/ $\overline{\text{PCSS}}$ —Peripheral Chip Select 5/Peripheral Chip Select Strobe

Master mode:

- Peripheral Chip Select 5 (O)—Used only when the peripheral-chip-select strobe is disabled (MCR[PCSSE]). Selects an DSPI slave to receive data transmitted by the module.
- Peripheral Chip Select Strobe (O)—Used only when the peripheral-chip-select strobe is enabled (MCR[PCSSE]). Strokes an off-module peripheral-chip-select demultiplexer, which decodes the module's PCS signals other than PCS5, preventing glitches on the demultiplexer outputs.

Slave mode: Unused

### 39.3.6 SCK—Serial Clock

Master mode: Serial Clock (O)—Supplies a clock signal from the module to DSPI slaves.

Slave mode: Serial Clock (I)—Supplies a clock signal to the module from an DSPI master.

### 39.3.7 SIN—Serial Input

Master mode: Serial Input (I)—Receives serial data.

Slave mode: Serial Input (I)—Receives serial data.

### 39.3.8 SOUT—Serial Output

Master mode: Serial Output (O)—Transmits serial data.

Slave mode: Serial Output (O)—Transmits serial data.

## 39.4 Memory Map/Register Definition

Register accesses to memory addresses that are reserved or undefined result in a transfer error. Any Write access to the POPR and RXFRn also results in a transfer error.

### NOTE

While the module is in the running state, do not write to these registers:

- PISRn
- DIMRn, DPIRn, and SSRn

#### DSPI memory map

| Address offset (hex) | Register name                                                             | Width (in bits) | Access | Reset value | Section/page                |
|----------------------|---------------------------------------------------------------------------|-----------------|--------|-------------|-----------------------------|
| 0                    | Module Configuration Register (DSPI_MCR)                                  | 32              | R/W    | 0000_0001h  | <a href="#">39.4.1/1876</a> |
| 8                    | Transfer Count Register (DSPI_TCR)                                        | 32              | R/W    | 0000_0000h  | <a href="#">39.4.2/1879</a> |
| C                    | Clock and Transfer Attributes Register (In Master Mode) (DSPI_CTAR0)      | 32              | R/W    | 7800_0000h  | <a href="#">39.4.3/1880</a> |
| C                    | Clock and Transfer Attributes Register (In Slave Mode) (DSPI_CTAR0_SLAVE) | 32              | R/W    | 7800_0000h  | <a href="#">39.4.4/1885</a> |
| 10                   | Clock and Transfer Attributes Register (In Master Mode) (DSPI_CTAR1)      | 32              | R/W    | 7800_0000h  | <a href="#">39.4.3/1880</a> |
| 10                   | Clock and Transfer Attributes Register (In Slave Mode) (DSPI_CTAR1_SLAVE) | 32              | R/W    | 7800_0000h  | <a href="#">39.4.4/1885</a> |
| 14                   | Clock and Transfer Attributes Register (In Master Mode) (DSPI_CTAR2)      | 32              | R/W    | 7800_0000h  | <a href="#">39.4.3/1880</a> |
| 18                   | Clock and Transfer Attributes Register (In Master Mode) (DSPI_CTAR3)      | 32              | R/W    | 7800_0000h  | <a href="#">39.4.3/1880</a> |
| 1C                   | Clock and Transfer Attributes Register (In Master Mode) (DSPI_CTAR4)      | 32              | R/W    | 7800_0000h  | <a href="#">39.4.3/1880</a> |
| 20                   | Clock and Transfer Attributes Register (In Master Mode) (DSPI_CTAR5)      | 32              | R/W    | 7800_0000h  | <a href="#">39.4.3/1880</a> |
| 24                   | Clock and Transfer Attributes Register (In Master Mode) (DSPI_CTAR6)      | 32              | R/W    | 7800_0000h  | <a href="#">39.4.3/1880</a> |
| 28                   | Clock and Transfer Attributes Register (In Master Mode) (DSPI_CTAR7)      | 32              | R/W    | 7800_0000h  | <a href="#">39.4.3/1880</a> |
| 2C                   | Status Register (DSPI_SR)                                                 | 32              | R/W    | 0201_0000h  | <a href="#">39.4.5/1887</a> |

Table continues on the next page...

## DSPI memory map (continued)

| Address offset (hex) | Register name                                                | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|--------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 30                   | DMA/Interrupt Request Select and Enable Register (DSPI_RSER) | 32              | R/W    | 0000_0000h  | <a href="#">39.4.6/1890</a>  |
| 34                   | PUSH TX FIFO Register In Master Mode (DSPI_PUSHR)            | 32              | R/W    | 0000_0000h  | <a href="#">39.4.7/1892</a>  |
| 34                   | PUSH TX FIFO Register In Slave Mode (DSPI_PUSHR_SLAVE)       | 32              | R/W    | 0000_0000h  | <a href="#">39.4.8/1894</a>  |
| 38                   | POP RX FIFO Register (DSPI_POPR)                             | 32              | R      | 0000_0000h  | <a href="#">39.4.9/1895</a>  |
| 3C                   | Transmit FIFO Registers (DSPI_TXFR0)                         | 32              | R      | 0000_0000h  | <a href="#">39.4.10/1895</a> |
| 40                   | Transmit FIFO Registers (DSPI_TXFR1)                         | 32              | R      | 0000_0000h  | <a href="#">39.4.10/1895</a> |
| 44                   | Transmit FIFO Registers (DSPI_TXFR2)                         | 32              | R      | 0000_0000h  | <a href="#">39.4.10/1895</a> |
| 48                   | Transmit FIFO Registers (DSPI_TXFR3)                         | 32              | R      | 0000_0000h  | <a href="#">39.4.10/1895</a> |
| 7C                   | Receive FIFO Registers (DSPI_RXFR0)                          | 32              | R      | 0000_0000h  | <a href="#">39.4.11/1896</a> |
| 80                   | Receive FIFO Registers (DSPI_RXFR1)                          | 32              | R      | 0000_0000h  | <a href="#">39.4.11/1896</a> |
| 84                   | Receive FIFO Registers (DSPI_RXFR2)                          | 32              | R      | 0000_0000h  | <a href="#">39.4.11/1896</a> |
| 88                   | Receive FIFO Registers (DSPI_RXFR3)                          | 32              | R      | 0000_0000h  | <a href="#">39.4.11/1896</a> |
| BC                   | DSI Configuration Register 0 (DSPI_DSICR0)                   | 32              | R/W    | 0000_0000h  | <a href="#">39.4.12/1897</a> |
| C0                   | DSI Serialization Data Register 0 (DSPI_SDR0)                | 32              | R      | 0000_0000h  | <a href="#">39.4.13/1899</a> |
| C4                   | DSI Alternate Serialization Data Register 0 (DSPI_ASDR0)     | 32              | R/W    | 0000_0000h  | <a href="#">39.4.14/1900</a> |
| C8                   | DSI Transmit Comparison Register 0 (DSPI_COMPR0)             | 32              | R      | 0000_0000h  | <a href="#">39.4.15/1900</a> |
| CC                   | DSI Deserialization Data Register 0 (DSPI_DDR0)              | 32              | R      | 0000_0000h  | <a href="#">39.4.16/1901</a> |
| D0                   | DSI Configuration Register 1 (DSPI_DSICR1)                   | 32              | R/W    | 0000_0000h  | <a href="#">39.4.17/1902</a> |
| D4                   | DSI Serialization Source Select Register 0 (DSPI_SSR0)       | 32              | R/W    | 0000_0000h  | <a href="#">39.4.18/1903</a> |
| D8                   | DSI Parallel Input Select Register 0 (DSPI_PISR0)            | 32              | R/W    | 0000_0000h  | <a href="#">39.4.19/1904</a> |
| DC                   | DSI Parallel Input Select Register 1 (DSPI_PISR1)            | 32              | R/W    | 0000_0000h  | <a href="#">39.4.20/1905</a> |
| E0                   | DSI Parallel Input Select Register 2 (DSPI_PISR2)            | 32              | R/W    | 0000_0000h  | <a href="#">39.4.21/1906</a> |
| E4                   | DSI Parallel Input Select Register 3 (DSPI_PISR3)            | 32              | R/W    | 0000_0000h  | <a href="#">39.4.22/1907</a> |

Table continues on the next page...

## DSPI memory map (continued)

| Address offset (hex) | Register name                                                    | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| E8                   | DSI Deserialized Data Interrupt Mask Register 0 (DSPI_DIMR0)     | 32              | R/W    | 0000_0000h  | <a href="#">39.4.23/1908</a> |
| EC                   | DSI Deserialized Data Polarity Interrupt Register 0 (DSPI_DPIR0) | 32              | R/W    | 0000_0000h  | <a href="#">39.4.24/1908</a> |

## 39.4.1 Module Configuration Register (DSPI\_MCR)

Contains bits to configure various attributes associated with the module operations. The HALT and MDIS bits can be changed at any time, but the effect takes place only on the next frame boundary. Only the HALT and MDIS bits in the MCR can be changed, while the module is in the Running state.

Address: 0h base + 0h offset = 0h

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
| R     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0  | 0  | 0  | 0  |    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |    |    |    |    | 0  | 0  |    |    |    |    | 0  |    |    |    |    |    |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1  |



## DSPI\_MCR field descriptions

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>MSTR       | <p>Master/Slave Mode Select</p> <p>Enables either Master mode (if supported) or Slave mode (if supported) operation.</p> <p>0 Enables Slave mode<br/>1 Enables Master mode</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 1<br>CONT_SCKE  | <p>Continuous SCK Enable</p> <p>Enables the Serial Communication Clock (SCK) to run continuously.</p> <p>0 Continuous SCK disabled.<br/>1 Continuous SCK enabled.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 2–3<br>DCONF    | <p>Interface Configuration</p> <p>Selects among the different interface configurations of the module.</p> <p>00 SPI<br/>01 DSI<br/>10 CSI<br/>11 Reserved</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 5<br>MTFE       | <p>Modified Transfer Format Enable</p> <p>Enables a modified transfer format to be used.</p> <p><b>NOTE:</b> When MTFE=1 with continuous SCK enabled (MCR [CONT_SCKE] =1) in master mode, configure CTAR[LSBFE]=0 for correct operations while receiving unequal length frames. If PUSHR[CONT] is also set for back to back frame transfer, also configure the frame size of the first frame as less than or equal to the frame size of the next frame. In this scenario, make sure that for all received frames, the bits are read equal to their respective frame sizes and any extra bits during POP operation are masked.</p> <p>0 Modified DSPI transfer format disabled.<br/>1 Modified DSPI transfer format enabled.</p> |
| 6<br>PCSSE      | <p>Peripheral Chip Select Strobe Enable</p> <p>Enables the PCS5/ <math>\overline{\text{PCSS}}</math> to operate as a PCS Strobe output signal.</p> <p>0 PCS5/ <math>\overline{\text{PCSS}}</math> is used as the Peripheral Chip Select[5] signal.<br/>1 PCS5/ <math>\overline{\text{PCSS}}</math> is used as an active-low PCS Strobe signal.</p>                                                                                                                                                                                                                                                                                                                                                                              |
| 7<br>ROOE       | <p>Receive FIFO Overflow Overwrite Enable</p> <p>In the RX FIFO overflow condition, configures the module to ignore the incoming serial data or overwrite existing data. If the RX FIFO is full and new data is received, the data from the transfer, generating the overflow, is ignored or shifted into the shift register.</p> <p>0 Incoming data is ignored.<br/>1 Incoming data is shifted into the shift register.</p>                                                                                                                                                                                                                                                                                                    |
| 8–9<br>Reserved | <p>Always write the reset value to this field.</p> <p>This field is reserved.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 10–15<br>PC SIS | Peripheral Chip Select x Inactive State                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

Table continues on the next page...

## DSPI\_MCR field descriptions (continued)

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                  | <p>Determines the inactive state of PCSx. Refer to the chip-specific D SPI information for the number of PCS signals used in this chip.</p> <p><b>NOTE:</b> The effect of this bit only takes place when module is enabled. Ensure that this bit is configured correctly before enabling the DSPI interface.</p> <p>0 The inactive state of PCSx is low.<br/>1 The inactive state of PCSx is high.</p>                                                                                                     |
| 16<br>Reserved   | This field is reserved.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 17<br>MDIS       | <p>Module Disable</p> <p>Allows the clock to be stopped to the non-memory mapped logic in the module effectively putting it in a software-controlled power-saving state. The reset value of the MDIS bit is parameterized, with a default reset value of 0. When the module is used in Slave Mode, it is recommended to leave this bit 0, because a slave doesn't have control over master transactions.</p> <p>0 Enables the module clocks.<br/>1 Allows external logic to disable the module clocks.</p> |
| 18<br>DIS_TXF    | <p>Disable Transmit FIFO</p> <p>When the TX FIFO is disabled, the transmit part of the module operates as a simplified double-buffered SPI. This bit can be written only when the MDIS bit is cleared.</p> <p>0 TX FIFO is enabled.<br/>1 TX FIFO is disabled.</p>                                                                                                                                                                                                                                         |
| 19<br>DIS_RXF    | <p>Disable Receive FIFO</p> <p>When the RX FIFO is disabled, the receive part of the module operates as a simplified double-buffered D SPI. This bit can only be written when the MDIS bit is cleared.</p> <p>0 RX FIFO is enabled.<br/>1 RX FIFO is disabled.</p>                                                                                                                                                                                                                                         |
| 20<br>CLR_TXF    | <p>Clear TX FIFO</p> <p>Flushes the TX FIFO. Writing a 1 to CLR_TXF clears the TX FIFO Counter. The CLR_TXF bit is always read as zero.</p> <p>0 Do not clear the TX FIFO counter.<br/>1 Clear the TX FIFO counter.</p>                                                                                                                                                                                                                                                                                    |
| 21<br>CLR_RXF    | <p>Clear RX FIFO</p> <p>Flushes the RX FIFO. Writing a 1 to CLR_RXF clears the RX Counter. The CLR_RXF bit is always read as zero.</p> <p>0 Do not clear the RX FIFO counter.<br/>1 Clear the RX FIFO counter.</p>                                                                                                                                                                                                                                                                                         |
| 22–23<br>SMPL_PT | <p>Sample Point</p> <p>Controls when the module master samples SIN in Modified Transfer Format. This field is valid only when CPHA bit in CTARn[CPHA] is 0.</p> <p>00 0 protocol clock cycles between SCK edge and SIN sample</p>                                                                                                                                                                                                                                                                          |

*Table continues on the next page...*

**DSPI\_MCR field descriptions (continued)**

| Field             | Description                                                                                                                                                                                  |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | 01 1 protocol clock cycle between SCK edge and SIN sample<br>10 2 protocol clock cycles between SCK edge and SIN sample<br>11 Reserved                                                       |
| 24–28<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                      |
| 29<br>Reserved    | This field is reserved.                                                                                                                                                                      |
| 30<br>PES         | Parity Error Stop<br><br>Controls D SPI operation when a parity error is detected in a received SPI frame.<br><br>0 D SPI frame transmission continues.<br>1 D SPI frame transmission stops. |
| 31<br>HALT        | Halt<br><br>The HALT bit starts and stops frame transfers. See <a href="#">Start and Stop of Module transfers</a><br><br>0 Start transfers.<br>1 Stop transfers.                             |

**39.4.2 Transfer Count Register (DSPI\_TCR)**

TCR contains a counter that indicates the number of DSPI transfers made. The transfer counter is intended to assist in queue management. Do not write the TCR when the module is in the Running state.

Address: 0h base + 8h offset = 8h

|       |           |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-----------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0         | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | DSPI_TCNT |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |           |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0         | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**DSPI\_TCR field descriptions**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>DSPI_TCNT | DSPI Transfer Counter<br><br>Counts the number of SPI transfers the module makes. The DSPI_TCNT field increments every time the last bit of an DSPI frame is transmitted. A value written to DSPI_TCNT presets the counter to that value. DSPI_TCNT is reset to zero at the beginning of the frame when the CTCNT field is set in the executing SPI command. The Transfer Counter wraps around; incrementing the counter past 65535 resets the counter to zero. |
| 16–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                         |

### 39.4.3 Clock and Transfer Attributes Register (In Master Mode) (DSPI\_CTAR<sub>n</sub>)

CTAR registers are used to define different transfer attributes. Do not write to the CTAR registers while the module is in the Running state.

In Master mode, the CTAR registers define combinations of transfer attributes such as frame size, clock phase and polarity, data bit ordering, baud rate, and various delays. In slave mode, a subset of the bitfields in CTAR0 and CTAR1 are used to set the slave transfer attributes.

When the module is configured as a DSPI master, the CTAS field in the command portion of the TX FIFO entry selects which of the CTAR registers is used. When the module is configured as an DSPI bus slave, it uses the CTAR0 register.

When the module is configured as a DSI master, the DSICTAS field in the DSI Configuration Register 0 (DSICR0), selects which of the CTAR register is used. When the module is configured as a DSI bus slave, the CTAR1 register is used.

In CSI Configuration, the transfer attributes are selected based on whether the current frame is DSPI data or DSI data. DSPI transfers in CSI Configuration follow the protocol described for DSPI Configuration, and DSI transfers in CSI Configuration follow the protocol described for DSI Configuration. CSI Configuration is valid only in conjunction with master mode.

TSB mode sets some limitations on transfer attributes:

- Clock phase is forced to be CPHA = 1 and the CPHA bit setting has no effect.
- PCS lines are driven at the driving edge of the SCK clock together with SOUT, so PCS assertion and negation delays control is unavailable and PCSSCK, PASC, CSSCK and ASC fields have no effect.
- Delay after transfer can be set from 1 to 64 serial clocks with help of PDT and DT fields

Address: 0h base + Ch offset + (4d × i), where i=0d to 7d

| Bit   | 0   | 1 | 2 | 3 | 4    | 5 | 6 | 7 | 8    | 9    | 10    | 11     | 12 | 13   | 14 | 15  |   |     |  |
|-------|-----|---|---|---|------|---|---|---|------|------|-------|--------|----|------|----|-----|---|-----|--|
| R     | DBR |   |   |   | FMSZ |   |   |   | CPOL | CPHA | LSBFE | PCSSCK |    | PASC |    | PDT |   | PBR |  |
| W     |     |   |   |   |      |   |   |   |      |      |       |        |    |      |    |     |   |     |  |
| Reset | 0   | 1 | 1 | 1 | 1    | 0 | 0 | 0 | 0    | 0    | 0     | 0      | 0  | 0    | 0  | 0   | 0 | 0   |  |

|       |       |    |    |    |     |    |    |    |    |    |    |    |    |    |    |    |
|-------|-------|----|----|----|-----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 16    | 17 | 18 | 19 | 20  | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | CSSCK |    |    |    | ASC |    |    |    | DT |    |    |    | BR |    |    |    |
| W     |       |    |    |    |     |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0     | 0  | 0  | 0  | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**DSPI\_CTAR<sub>n</sub> field descriptions**

| Field       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |     |                |     |                |   |     |     |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|----------------|-----|----------------|---|-----|-----|-------|---|---|----|-------|---|---|----|-------|---|---|----|-------|---|---|----|-------|---|---|----|-------|---|---|----|-------|---|---|----|-------|---|---|----|-------|
| 0<br>DBR    | <p>Double Baud Rate</p> <p>Doubles the effective baud rate of the Serial Communications Clock (SCK). This field is used only in master mode. It effectively halves the Baud Rate division ratio, supporting faster frequencies, and odd division ratios for the Serial Communications Clock (SCK). When the DBR bit is set, the duty cycle of the Serial Communications Clock (SCK) depends on the value in the Baud Rate Prescaler and the Clock Phase bit as listed in the following table. See the BR field description for details on how to compute the baud rate.</p> <p style="text-align: center;"><b>Table 39-15. DSPI SCK Duty Cycle</b></p> <table><tr><th>DBR</th><th>CPHA</th><th>PBR</th><th>SCK Duty Cycle</th></tr><tr><td>0</td><td>any</td><td>any</td><td>50/50</td></tr><tr><td>1</td><td>0</td><td>00</td><td>50/50</td></tr><tr><td>1</td><td>0</td><td>01</td><td>33/66</td></tr><tr><td>1</td><td>0</td><td>10</td><td>40/60</td></tr><tr><td>1</td><td>0</td><td>11</td><td>43/57</td></tr><tr><td>1</td><td>1</td><td>00</td><td>50/50</td></tr><tr><td>1</td><td>1</td><td>01</td><td>66/33</td></tr><tr><td>1</td><td>1</td><td>10</td><td>60/40</td></tr><tr><td>1</td><td>1</td><td>11</td><td>57/43</td></tr></table> <p>0 The baud rate is computed normally with a 50/50 duty cycle.</p> <p>1 The baud rate is doubled with the duty cycle depending on the Baud Rate Prescaler.</p> | DBR | CPHA           | PBR | SCK Duty Cycle | 0 | any | any | 50/50 | 1 | 0 | 00 | 50/50 | 1 | 0 | 01 | 33/66 | 1 | 0 | 10 | 40/60 | 1 | 0 | 11 | 43/57 | 1 | 1 | 00 | 50/50 | 1 | 1 | 01 | 66/33 | 1 | 1 | 10 | 60/40 | 1 | 1 | 11 | 57/43 |
| DBR         | CPHA                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | PBR | SCK Duty Cycle |     |                |   |     |     |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |
| 0           | any                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | any | 50/50          |     |                |   |     |     |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |
| 1           | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 00  | 50/50          |     |                |   |     |     |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |
| 1           | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 01  | 33/66          |     |                |   |     |     |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |
| 1           | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 10  | 40/60          |     |                |   |     |     |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |
| 1           | 0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 11  | 43/57          |     |                |   |     |     |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |
| 1           | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 00  | 50/50          |     |                |   |     |     |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |
| 1           | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 01  | 66/33          |     |                |   |     |     |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |
| 1           | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 10  | 60/40          |     |                |   |     |     |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |
| 1           | 1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 11  | 57/43          |     |                |   |     |     |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |
| 1–4<br>FMSZ | <p>Frame Size</p> <p>The number of bits transferred per frame is equal to the FMSZ value plus 1. Regardless of the transmission mode, the minimum valid frame size value is 4. There is a constraint on the minimum allowable Frame size, which depends on the ratio of the Register Read/Write clock to the Protocol clock. The minimum Frame size (FMSZ + 1) is determined by the following equation. Upper Ceiling must be applied for non-integer values.</p> <p>Minimum Frame Size = ( ( 4 x f<sub>p</sub> ) + ( 3 x f<sub>r</sub> ) ) / ( n x f<sub>r</sub> )</p> <p>f<sub>p</sub> = Protocol clock frequency</p> <p>f<sub>r</sub> = Register interface clock frequency</p> <p>n = (PBR x BR) / ( 1 + DBR) = Multiple of protocol clock required to get Baud (SCK) Clock</p> <p>However, the minimum frame size can never be less than 4. There is no constraint on the maximum programmable frame size. Also note that 'fp' can be equal to, less than or greater than 'fr' purely based on user requirement.</p>                                                                                                                                                                                                                                                                                                                                                                                              |     |                |     |                |   |     |     |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |   |   |    |       |

Table continues on the next page...

DSPI\_CTAR<sub>n</sub> field descriptions (continued)

| Field         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 5<br>CPOL     | <p>Clock Polarity</p> <p>Selects the inactive state of the Serial Communications Clock (SCK). This bit is used in both master and slave mode. For successful communication between serial devices, the devices must have identical clock polarities. When the Continuous Selection Format is selected, switching between clock polarities without stopping the module can cause errors in the transfer due to the peripheral device interpreting the switch of clock polarity as a valid clock edge.</p> <p><b>NOTE:</b> In case of Continuous SCK mode, when the module goes in low power mode(disabled), inactive state of SCK is not guaranteed.</p> <p>0 The inactive state value of SCK is low.<br/>1 The inactive state value of SCK is high.</p> |
| 6<br>CPHA     | <p>Clock Phase</p> <p>Selects which edge of SCK causes data to change and which edge causes data to be captured. This bit is used in both master and slave mode. For successful communication between serial devices, the devices must have identical clock phase settings. In Continuous SCK mode or TSB mode, the bit value is ignored and the transfers are done as if the CPHA bit is set to 1.</p> <p>0 Data is captured on the leading edge of SCK and changed on the following edge.<br/>1 Data is changed on the leading edge of SCK and captured on the following edge.</p>                                                                                                                                                                    |
| 7<br>LSBFE    | <p>LSB First</p> <p>Specifies whether the LSB or MSB of the frame is transferred first.</p> <p>0 Data is transferred MSB first.<br/>1 Data is transferred LSB first.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 8–9<br>PCSSCK | <p>PCS to SCK Delay Prescaler</p> <p>Selects the prescaler value for the delay between assertion of PCS and the first edge of the SCK. See the CSSCK field description for information on how to compute the PCS to SCK Delay. Refer <a href="#">PCS to SCK Delay (t<sub>CSC</sub>)</a> for more details.</p> <p>00 PCS to SCK Prescaler value is 1.<br/>01 PCS to SCK Prescaler value is 3.<br/>10 PCS to SCK Prescaler value is 5.<br/>11 PCS to SCK Prescaler value is 7.</p>                                                                                                                                                                                                                                                                        |
| 10–11<br>PASC | <p>After SCK Delay Prescaler</p> <p>Selects the prescaler value for the delay between the last edge of SCK and the negation of PCS. See the ASC field description for information on how to compute the After SCK Delay. In the TSB mode, the PASC field has no effect. Refer <a href="#">After SCK Delay (t<sub>ASC</sub>)</a> for more details.</p> <p>00 Delay after Transfer Prescaler value is 1.<br/>01 Delay after Transfer Prescaler value is 3.<br/>10 Delay after Transfer Prescaler value is 5.<br/>11 Delay after Transfer Prescaler value is 7.</p>                                                                                                                                                                                        |
| 12–13<br>PDT  | <p>Delay after Transfer Prescaler</p> <p>Selects the prescaler value for the delay between the negation of the PCS signal at the end of a frame and the assertion of PCS at the beginning of the next frame. The PDT field is only used in master mode. See</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

*Table continues on the next page...*

DSPI\_CTAR<sub>n</sub> field descriptions (continued)

| Field          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |             |                    |      |   |      |   |      |   |      |    |      |    |      |    |      |     |      |     |      |     |      |      |      |      |      |      |      |      |      |       |      |       |      |       |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|--------------------|------|---|------|---|------|---|------|----|------|----|------|----|------|-----|------|-----|------|-----|------|------|------|------|------|------|------|------|------|-------|------|-------|------|-------|
|                | <p>the DT field description for details on how to compute the Delay after Transfer. Refer <a href="#">Delay after Transfer (t<sub>DT</sub>)</a> for more details.</p> <p>00 Delay after Transfer Prescaler value is 1.<br/> 01 Delay after Transfer Prescaler value is 3.<br/> 10 Delay after Transfer Prescaler value is 5.<br/> 11 Delay after Transfer Prescaler value is 7.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |             |                    |      |   |      |   |      |   |      |    |      |    |      |    |      |     |      |     |      |     |      |      |      |      |      |      |      |      |      |       |      |       |      |       |
| 14–15<br>PBR   | <p>Baud Rate Prescaler</p> <p>Selects the prescaler value for the baud rate. This field is used only in master mode. The baud rate is the frequency of the SCK. The protocol clock is divided by the prescaler value before the baud rate selection takes place. See the BR field description for details on how to compute the baud rate.</p> <p>00 Baud Rate Prescaler value is 2.<br/> 01 Baud Rate Prescaler value is 3.<br/> 10 Baud Rate Prescaler value is 5.<br/> 11 Baud Rate Prescaler value is 7.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |             |                    |      |   |      |   |      |   |      |    |      |    |      |    |      |     |      |     |      |     |      |      |      |      |      |      |      |      |      |       |      |       |      |       |
| 16–19<br>CSSCK | <p>PCS to SCK Delay Scaler</p> <p>Selects the scaler value for the PCS to SCK delay. This field is used only in master mode. In the TSB mode the field has no effect. The PCS to SCK Delay is the delay between the assertion of PCS and the first edge of the SCK. The delay is a multiple of the protocol clock period, and it is computed according to the following equation:</p> $t_{CSC} = (1/f_P) \times PCSSCK \times CSSCK.$ <p>The following table lists the delay scaler values.</p> <p style="text-align: center;"><b>Table 39-14. Delay Scaler Encoding</b></p> <table> <tr> <th>Field Value</th><th>Delay Scaler Value</th></tr> <tr><td>0000</td><td>2</td></tr> <tr><td>0001</td><td>4</td></tr> <tr><td>0010</td><td>8</td></tr> <tr><td>0011</td><td>16</td></tr> <tr><td>0100</td><td>32</td></tr> <tr><td>0101</td><td>64</td></tr> <tr><td>0110</td><td>128</td></tr> <tr><td>0111</td><td>256</td></tr> <tr><td>1000</td><td>512</td></tr> <tr><td>1001</td><td>1024</td></tr> <tr><td>1010</td><td>2048</td></tr> <tr><td>1011</td><td>4096</td></tr> <tr><td>1100</td><td>8192</td></tr> <tr><td>1101</td><td>16384</td></tr> <tr><td>1110</td><td>32768</td></tr> <tr><td>1111</td><td>65536</td></tr> </table> | Field Value | Delay Scaler Value | 0000 | 2 | 0001 | 4 | 0010 | 8 | 0011 | 16 | 0100 | 32 | 0101 | 64 | 0110 | 128 | 0111 | 256 | 1000 | 512 | 1001 | 1024 | 1010 | 2048 | 1011 | 4096 | 1100 | 8192 | 1101 | 16384 | 1110 | 32768 | 1111 | 65536 |
| Field Value    | Delay Scaler Value                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |             |                    |      |   |      |   |      |   |      |    |      |    |      |    |      |     |      |     |      |     |      |      |      |      |      |      |      |      |      |       |      |       |      |       |
| 0000           | 2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |             |                    |      |   |      |   |      |   |      |    |      |    |      |    |      |     |      |     |      |     |      |      |      |      |      |      |      |      |      |       |      |       |      |       |
| 0001           | 4                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |             |                    |      |   |      |   |      |   |      |    |      |    |      |    |      |     |      |     |      |     |      |      |      |      |      |      |      |      |      |       |      |       |      |       |
| 0010           | 8                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |             |                    |      |   |      |   |      |   |      |    |      |    |      |    |      |     |      |     |      |     |      |      |      |      |      |      |      |      |      |       |      |       |      |       |
| 0011           | 16                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |             |                    |      |   |      |   |      |   |      |    |      |    |      |    |      |     |      |     |      |     |      |      |      |      |      |      |      |      |      |       |      |       |      |       |
| 0100           | 32                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |             |                    |      |   |      |   |      |   |      |    |      |    |      |    |      |     |      |     |      |     |      |      |      |      |      |      |      |      |      |       |      |       |      |       |
| 0101           | 64                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |             |                    |      |   |      |   |      |   |      |    |      |    |      |    |      |     |      |     |      |     |      |      |      |      |      |      |      |      |      |       |      |       |      |       |
| 0110           | 128                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |             |                    |      |   |      |   |      |   |      |    |      |    |      |    |      |     |      |     |      |     |      |      |      |      |      |      |      |      |      |       |      |       |      |       |
| 0111           | 256                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |             |                    |      |   |      |   |      |   |      |    |      |    |      |    |      |     |      |     |      |     |      |      |      |      |      |      |      |      |      |       |      |       |      |       |
| 1000           | 512                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |             |                    |      |   |      |   |      |   |      |    |      |    |      |    |      |     |      |     |      |     |      |      |      |      |      |      |      |      |      |       |      |       |      |       |
| 1001           | 1024                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |             |                    |      |   |      |   |      |   |      |    |      |    |      |    |      |     |      |     |      |     |      |      |      |      |      |      |      |      |      |       |      |       |      |       |
| 1010           | 2048                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |             |                    |      |   |      |   |      |   |      |    |      |    |      |    |      |     |      |     |      |     |      |      |      |      |      |      |      |      |      |       |      |       |      |       |
| 1011           | 4096                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |             |                    |      |   |      |   |      |   |      |    |      |    |      |    |      |     |      |     |      |     |      |      |      |      |      |      |      |      |      |       |      |       |      |       |
| 1100           | 8192                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |             |                    |      |   |      |   |      |   |      |    |      |    |      |    |      |     |      |     |      |     |      |      |      |      |      |      |      |      |      |       |      |       |      |       |
| 1101           | 16384                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |             |                    |      |   |      |   |      |   |      |    |      |    |      |    |      |     |      |     |      |     |      |      |      |      |      |      |      |      |      |       |      |       |      |       |
| 1110           | 32768                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |             |                    |      |   |      |   |      |   |      |    |      |    |      |    |      |     |      |     |      |     |      |      |      |      |      |      |      |      |      |       |      |       |      |       |
| 1111           | 65536                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |             |                    |      |   |      |   |      |   |      |    |      |    |      |    |      |     |      |     |      |     |      |      |      |      |      |      |      |      |      |       |      |       |      |       |

Table continues on the next page...

## DSPI\_CTARn field descriptions (continued)

| Field        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |           |                        |      |   |      |   |      |   |      |   |      |    |      |    |      |    |      |     |      |     |      |     |      |      |      |      |      |      |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------|------------------------|------|---|------|---|------|---|------|---|------|----|------|----|------|----|------|-----|------|-----|------|-----|------|------|------|------|------|------|
|              | Refer <a href="#">PCS to SCK Delay (t<sub>CSC</sub>)</a> for more details.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |           |                        |      |   |      |   |      |   |      |   |      |    |      |    |      |    |      |     |      |     |      |     |      |      |      |      |      |      |
| 20–23<br>ASC | <p>After SCK Delay Scaler</p> <p>Selects the scaler value for the After SCK Delay. This field is used only in master mode. The After SCK Delay is the delay between the last edge of SCK and the negation of PCS. The delay is a multiple of the protocol clock period, and it is computed according to the following equation:</p> $t_{ASC} = (1/f_P) \times PASC \times ASC$ <p>See Delay Scaler Encoding table in CTARn[CSSCK] bit field description for scaler values. Refer <a href="#">After SCK Delay (t<sub>ASC</sub>)</a> for more details.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |           |                        |      |   |      |   |      |   |      |   |      |    |      |    |      |    |      |     |      |     |      |     |      |      |      |      |      |      |
| 24–27<br>DT  | <p>Delay After Transfer Scaler</p> <p>Selects the Delay after Transfer Scaler. This field is used only in master mode. The Delay after Transfer is the time between the negation of the PCS signal at the end of a frame and the assertion of PCS at the beginning of the next frame.</p> <p>In the Continuous Serial Communications Clock operation, the DT value is fixed to one SCK clock period, The Delay after Transfer is a multiple of the protocol clock period, and it is computed according to the following equation:</p> $t_{DT} = (1/f_P) \times PDT \times DT$ <p>See Delay Scaler Encoding table in CTARn[CSSCK] bit field description for scaler values.</p> <p>In the TSB mode, the Delay after Transfer is equal to a number formed by concatenation of the PDT and DT fields plus 1 of the SCK clock periods. Refer <a href="#">Delay after Transfer (t<sub>DT</sub>)</a> for more details.</p>                                                                                                                 |           |                        |      |   |      |   |      |   |      |   |      |    |      |    |      |    |      |     |      |     |      |     |      |      |      |      |      |      |
| 28–31<br>BR  | <p>Baud Rate Scaler</p> <p>Selects the scaler value for the baud rate. This field is used only in master mode. The prescaled protocol clock is divided by the Baud Rate Scaler to generate the frequency of the SCK. The baud rate is computed according to the following equation:</p> $SCK \text{ baud rate} = (f_P / PBR) \times [(1+DBR)/BR]$ <p>The following table lists the baud rate scaler values.</p> <p style="text-align: center;"><b>Table 39-13. Baud Rate Scaler</b></p> <table> <tr> <th>CTARn[BR]</th><th>Baud Rate Scaler Value</th></tr> <tr><td>0000</td><td>2</td></tr> <tr><td>0001</td><td>4</td></tr> <tr><td>0010</td><td>6</td></tr> <tr><td>0011</td><td>8</td></tr> <tr><td>0100</td><td>16</td></tr> <tr><td>0101</td><td>32</td></tr> <tr><td>0110</td><td>64</td></tr> <tr><td>0111</td><td>128</td></tr> <tr><td>1000</td><td>256</td></tr> <tr><td>1001</td><td>512</td></tr> <tr><td>1010</td><td>1024</td></tr> <tr><td>1011</td><td>2048</td></tr> <tr><td>1100</td><td>4096</td></tr> </table> | CTARn[BR] | Baud Rate Scaler Value | 0000 | 2 | 0001 | 4 | 0010 | 6 | 0011 | 8 | 0100 | 16 | 0101 | 32 | 0110 | 64 | 0111 | 128 | 1000 | 256 | 1001 | 512 | 1010 | 1024 | 1011 | 2048 | 1100 | 4096 |
| CTARn[BR]    | Baud Rate Scaler Value                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |           |                        |      |   |      |   |      |   |      |   |      |    |      |    |      |    |      |     |      |     |      |     |      |      |      |      |      |      |
| 0000         | 2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |           |                        |      |   |      |   |      |   |      |   |      |    |      |    |      |    |      |     |      |     |      |     |      |      |      |      |      |      |
| 0001         | 4                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |           |                        |      |   |      |   |      |   |      |   |      |    |      |    |      |    |      |     |      |     |      |     |      |      |      |      |      |      |
| 0010         | 6                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |           |                        |      |   |      |   |      |   |      |   |      |    |      |    |      |    |      |     |      |     |      |     |      |      |      |      |      |      |
| 0011         | 8                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |           |                        |      |   |      |   |      |   |      |   |      |    |      |    |      |    |      |     |      |     |      |     |      |      |      |      |      |      |
| 0100         | 16                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |           |                        |      |   |      |   |      |   |      |   |      |    |      |    |      |    |      |     |      |     |      |     |      |      |      |      |      |      |
| 0101         | 32                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |           |                        |      |   |      |   |      |   |      |   |      |    |      |    |      |    |      |     |      |     |      |     |      |      |      |      |      |      |
| 0110         | 64                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |           |                        |      |   |      |   |      |   |      |   |      |    |      |    |      |    |      |     |      |     |      |     |      |      |      |      |      |      |
| 0111         | 128                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |           |                        |      |   |      |   |      |   |      |   |      |    |      |    |      |    |      |     |      |     |      |     |      |      |      |      |      |      |
| 1000         | 256                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |           |                        |      |   |      |   |      |   |      |   |      |    |      |    |      |    |      |     |      |     |      |     |      |      |      |      |      |      |
| 1001         | 512                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |           |                        |      |   |      |   |      |   |      |   |      |    |      |    |      |    |      |     |      |     |      |     |      |      |      |      |      |      |
| 1010         | 1024                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |           |                        |      |   |      |   |      |   |      |   |      |    |      |    |      |    |      |     |      |     |      |     |      |      |      |      |      |      |
| 1011         | 2048                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |           |                        |      |   |      |   |      |   |      |   |      |    |      |    |      |    |      |     |      |     |      |     |      |      |      |      |      |      |
| 1100         | 4096                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |           |                        |      |   |      |   |      |   |      |   |      |    |      |    |      |    |      |     |      |     |      |     |      |      |      |      |      |      |

Table continues on the next page...



DSPI\_CTAR<sub>n</sub> field descriptions (continued)

| Field                                            | Description            |
|--------------------------------------------------|------------------------|
| <b>Table 39-13. Baud Rate Scaler (continued)</b> |                        |
| CTAR <sub>n</sub> [BR]                           | Baud Rate Scaler Value |
| 1101                                             | 8192                   |
| 1110                                             | 16384                  |
| 1111                                             | 32768                  |

### 39.4.4 Clock and Transfer Attributes Register (In Slave Mode) (DSPI\_CTAR<sub>n</sub>\_SLAVE)

When the module is configured as an SPI bus slave, the CTAR0 register is used.

When the module is configured as a DSI master, the DSICTAS field in the DSI Configuration Register 0 (DSICR0), selects which of the CTAR register is used. When the module is configured as a DSI bus slave, the CTAR1 register is used.

Address: 0h base + Ch offset + (4d × i), where i=0d to 1d

|       |          |    |    |    |    |      |      |    |    |          |          |    |    |    |    |    |
|-------|----------|----|----|----|----|------|------|----|----|----------|----------|----|----|----|----|----|
| Bit   | 0        | 1  | 2  | 3  | 4  | 5    | 6    | 7  | 8  | 9        | 10       | 11 | 12 | 13 | 14 | 15 |
| R     | FMSZ     |    |    |    |    | CPOL | CPHA | PE | PP | Reserved | Reserved |    |    |    |    |    |
| W     |          |    |    |    |    |      |      |    |    |          |          |    |    |    |    |    |
| Reset | 0        | 1  | 1  | 1  | 1  | 0    | 0    | 0  | 0  | 0        | 0        | 0  | 0  | 0  | 0  | 0  |
| Bit   | 16       | 17 | 18 | 19 | 20 | 21   | 22   | 23 | 24 | 25       | 26       | 27 | 28 | 29 | 30 | 31 |
| R     | Reserved |    |    |    |    |      |      |    |    |          |          |    |    |    |    |    |
| W     |          |    |    |    |    |      |      |    |    |          |          |    |    |    |    |    |
| Reset | 0        | 0  | 0  | 0  | 0  | 0    | 0    | 0  | 0  | 0        | 0        | 0  | 0  | 0  | 0  | 0  |

### DSPI\_CTAR<sub>n</sub>\_SLAVE field descriptions

| Field       | Description                                                                                                                                               |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–4<br>FMSZ | Frame Size<br><br>The number of bits transferred per frame is equal to the FMSZ field value plus 1. Note that the minimum valid value of frame size is 4. |
| 5<br>CPOL   | Clock Polarity<br><br>Selects the inactive state of the Serial Communications Clock (SCK).                                                                |

*Table continues on the next page...*

**DSPI\_CTARn\_SLAVE field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | <p><b>NOTE:</b> In case of Continuous SCK mode, when the module goes in low power mode(disabled), inactive state of SCK is not guaranteed.</p> <p>0 The inactive state value of SCK is low.</p> <p>1 The inactive state value of SCK is high.</p>                                                                                                                                                                                                                                                                                                                                       |
| 6<br>CPHA         | <p>Clock Phase</p> <p>Selects which edge of SCK causes data to change and which edge causes data to be captured. This bit is used in both master and slave mode. For successful communication between serial devices, the devices must have identical clock phase settings. In Continuous SCK mode or TSB mode, the bit value is ignored and the transfers are done as if the CPHA bit is set to 1.</p> <p>0 Data is captured on the leading edge of SCK and changed on the following edge.</p> <p>1 Data is changed on the leading edge of SCK and captured on the following edge.</p> |
| 7<br>PE           | <p>Parity Enable</p> <p>Enables parity bit transmission and reception for the frame.</p> <p>0 No parity bit included/checked.</p> <p>1 Parity bit is transmitted instead of last data bit in frame, parity checked for received frame.</p>                                                                                                                                                                                                                                                                                                                                              |
| 8<br>PP           | <p>Parity Polarity</p> <p>Controls polarity of the parity bit transmitted and checked.</p> <p>0 Even Parity: the number of 1 bits in the transmitted frame is even. The SR[SPEF] bit is set if the number of 1 bits is odd in the received frame.</p> <p>1 Odd Parity: the number of 1 bits in the transmitted frame is odd. The SR[SPEF] bit is set if the number of 1 bits is even in the received frame.</p>                                                                                                                                                                         |
| 9<br>Reserved     | This field is reserved.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 10–31<br>Reserved | This field is reserved.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

### 39.4.5 Status Register (DSPI\_SR)

SR contains status and flag bits. The bits reflect the status of the module and indicate the occurrence of events that can generate interrupt or DMA requests. Software can clear flag bits in the SR by writing a 1 to them. Writing a 0 to a flag bit has no effect. This register may not be writable in Module Disable mode due to the use of power saving mechanisms.

Address: 0h base + 2Ch offset = 2Ch

| Bit   | 0   | 1     | 2        | 3    | 4    | 5        | 6    | 7 | 8 | 9    | 10   | 11   | 12   | 13 | 14   | 15       |
|-------|-----|-------|----------|------|------|----------|------|---|---|------|------|------|------|----|------|----------|
| R     | TCF | TXRXS | Reserved | EOQF | TFUF | Reserved | TFFF | 0 | 0 | DPEF | SPEF | DDIF | RFOF | 0  | RFDF | Reserved |
| W     | w1c |       | w1c      | w1c  | w1c  | w1c      | w1c  |   |   | w1c  | w1c  | w1c  | w1c  |    | w1c  | w1c      |
| Reset | 0   | 0     | 0        | 0    | 0    | 0        | 1    | 0 | 0 | 0    | 0    | 0    | 0    | 0  | 0    | 1        |

| Bit   | 16    | 17 | 18 | 19 | 20       | 21 | 22 | 23 | 24    | 25 | 26 | 27 | 28        | 29 | 30 | 31 |
|-------|-------|----|----|----|----------|----|----|----|-------|----|----|----|-----------|----|----|----|
| R     | TXCTR |    |    |    | TXNXTPTR |    |    |    | RXCTR |    |    |    | POPNTPTTR |    |    |    |
| W     |       |    |    |    |          |    |    |    |       |    |    |    |           |    |    |    |
| Reset | 0     | 0  | 0  | 0  | 0        | 0  | 0  | 0  | 0     | 0  | 0  | 0  | 0         | 0  | 0  | 0  |

**DSPI\_SR field descriptions**

| Field    | Description                                                                                                                                                                                               |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>TCF | Transfer Complete Flag<br><br>Indicates that all bits in a frame have been shifted out. TCF remains set until it is cleared by writing a 1 to it.<br><br>0 Transfer not complete.<br>1 Transfer complete. |

*Table continues on the next page...*

## DSPI\_SR field descriptions (continued)

| Field         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1<br>TXRXS    | <p>TX and RX Status</p> <p>Reflects the run status of the module.</p> <p>0 Transmit and receive operations are disabled (The module is in Stopped state).</p> <p>1 Transmit and receive operations are enabled (The module is in Running state).</p>                                                                                                                                                                                                                                                                                                                                                |
| 2<br>Reserved | This field is reserved.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 3<br>EOQF     | <p>End of Queue Flag</p> <p>Indicates that the last entry in a queue has been transmitted when the module is in Master mode. The EOQF bit is set when the TX FIFO entry has the EOQ bit set in the command halfword and the end of the transfer is reached. The EOQF bit remains set until cleared by writing a 1 to it. When the EOQF bit is set, the TXRXS bit is automatically cleared.</p> <p>0 EOQ is not set in the executing command.</p> <p>1 EOQ is set in the executing SPI command.</p>                                                                                                  |
| 4<br>TFUF     | <p>Transmit FIFO Underflow Flag</p> <p>Indicates an underflow condition in the TX FIFO. The transmit underflow condition is detected only for DSPI blocks operating in Slave mode and DSPI configuration. TFUF is set when the TX FIFO of the module operating in SPI Slave mode is empty and an external DSPI master initiates a transfer. The TFUF bit remains set until cleared by writing 1 to it.</p> <p>0 No TX FIFO underflow.</p> <p>1 TX FIFO underflow has occurred.</p>                                                                                                                  |
| 5<br>Reserved | This field is reserved.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 6<br>TFFF     | <p>Transmit FIFO Fill Flag</p> <p>Indicates whether there is an available location to be filled in the FIFO. Either a DMA request or an interrupt indication can be used to add another entry to the FIFO. Note that this bit is set if at least one location is free in the FIFO. The TFFF bit can be cleared by writing 1 to it or by acknowledgement from the DMA controller to the TX FIFO full request, when the TX FIFO is full.</p> <p><b>NOTE:</b> The reset value of this bit is 0 when the module is disabled, (MCR[MDIS]=1).</p> <p>0 TX FIFO is full.</p> <p>1 TX FIFO is not full.</p> |
| 7<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 8<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 9<br>DPEF     | <p>DSI Parity Error Flag</p> <p>Indicates that a DSI frame with parity error had been received. The bit remains set until it is cleared by writing a 1 to it.</p> <p>0 No parity error.</p> <p>1 Parity error has occurred.</p>                                                                                                                                                                                                                                                                                                                                                                     |
| 10<br>SPEF    | <p>D</p> <p>SPI Parity Error Flag</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |

Table continues on the next page...

**DSPI\_SR field descriptions (continued)**

| <b>Field</b>       | <b>Description</b>                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                    | Indicates that a DSPI frame with parity error had been received. The bit remains set until it is cleared by writing a 1 to it.<br><br>0 No parity error.<br>1 Parity error has occurred.                                                                                                                                                                                                                                                                    |
| 11<br>DDIF         | DSI Data Received with Active Bits<br><br>Indicates that a DSI frame had been received with bits selected by DIMR with active polarity, defined by the DPIR register. The bit remains set until it is cleared by writing a 1 to it.<br><br>0 No DSI data with active bits was received.<br>1 DSI data with active bits was received.                                                                                                                        |
| 12<br>RFOF         | Receive FIFO Overflow Flag<br><br>Indicates an overflow condition in the RX FIFO. The field is set when the RX FIFO and shift register are full and a transfer is initiated. The bit remains set until it is cleared by writing a 1 to it.<br><br>0 No Rx FIFO overflow.<br>1 Rx FIFO overflow has occurred.                                                                                                                                                |
| 13<br>Reserved     | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                     |
| 14<br>RFDF         | Receive FIFO Drain Flag<br><br>Indicates whether there is an available location to be drained from the FIFO. Either a DMA request or an interrupt indication can be used to read from the FIFO. Note that this bit is set if at least one location can be read from the FIFO. The RFDF bit can be cleared by writing 1 to it or by acknowledgement from the DMA controller when the RX FIFO is empty.<br><br>0 RX FIFO is empty.<br>1 RX FIFO is not empty. |
| 15<br>Reserved     | This field is reserved.                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 16–19<br>TXCTR     | TX FIFO Counter<br><br>Indicates the number of valid entries in the TX FIFO. The TXCTR is incremented every time the PUSHHR is written. The TXCTR is decremented every time an DSPI command is executed and the DSPI data is transferred to the shift register.                                                                                                                                                                                             |
| 20–23<br>TXNXTPTR  | Transmit Next Pointer<br><br>Indicates which TX FIFO entry is transmitted during the next transfer. The TXNXTPTR field is updated every time SPI data is transferred from the TX FIFO to the shift register.                                                                                                                                                                                                                                                |
| 24–27<br>RXCTR     | RX FIFO Counter<br><br>Indicates the number of entries in the RX FIFO. The RXCTR is decremented every time the POPR is read. The RXCTR is incremented every time data is transferred from the shift register to the RX FIFO.                                                                                                                                                                                                                                |
| 28–31<br>POPNXTPTR | Pop Next Pointer<br><br>Contains a pointer to the RX FIFO entry to be returned when the POPR is read. The POPNXTPTR is updated when the POPR is read.                                                                                                                                                                                                                                                                                                       |

### 39.4.6 DMA/Interrupt Request Select and Enable Register (DSPI\_RSER)

RSER controls DMA and interrupt requests. Do not write to the RSER while the module is in the Running state.

Address: 0h base + 30h offset = 30h

| Bit   | 0      | 1        | 2        | 3       | 4       | 5        | 6       | 7         | 8        | 9       | 10      | 11      | 12      | 13       | 14      | 15        |
|-------|--------|----------|----------|---------|---------|----------|---------|-----------|----------|---------|---------|---------|---------|----------|---------|-----------|
| R     | TCF_RE | Reserved | Reserved | EOQF_RE | TFUF_RE | Reserved | TFFF_RE | TFFF_DIRS | Reserved | DPEF_RE | SPEF_RE | DDIF_RE | RFOF_RE | Reserved | RFDF_RE | RFDF_DIRS |
| W     |        |          |          |         |         |          |         |           |          |         |         |         |         |          |         |           |
| Reset | 0      | 0        | 0        | 0       | 0       | 0        | 0       | 0         | 0        | 0       | 0       | 0       | 0       | 0        | 0       | 0         |

| Bit   | 16       | 17       | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|-------|----------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| R     | Reserved | Reserved | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |          |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0        | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**DSPI\_RSER field descriptions**

| Field         | Description                                                                                                                                                                                     |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>TCF_RE   | Transmission Complete Request Enable<br><br>Enables TCF flag in the SR to generate an interrupt request.<br><br>0 TCF interrupt requests are disabled.<br>1 TCF interrupt requests are enabled. |
| 1<br>Reserved | Always write the reset value to this field.<br><br>This field is reserved.                                                                                                                      |
| 2<br>Reserved | This field is reserved.                                                                                                                                                                         |
| 3<br>EOQF_RE  | Finished Request Enable<br><br>Enables the EOQF flag in the SR to generate an interrupt request.<br><br>0 EOQF interrupt requests are disabled.<br>1 EOQF interrupt requests are enabled.       |
| 4<br>TFUF_RE  | Transmit FIFO Underflow Request Enable<br><br>Enables the TFUF flag in the SR to generate an interrupt request.                                                                                 |

*Table continues on the next page...*

**DSPI\_RSER field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                                                                                               |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | 0 TFUF interrupt requests are disabled.<br>1 TFUF interrupt requests are enabled.                                                                                                                                                                                                                                                         |
| 5<br>Reserved  | This field is reserved.                                                                                                                                                                                                                                                                                                                   |
| 6<br>TFFF_RE   | Transmit FIFO Fill Request Enable<br><br>Enables the TFFF flag in the SR to generate a request. The TFFF_DIRS bit selects between generating an interrupt request or a DMA request.<br><br>0 TFFF interrupts or DMA requests are disabled.<br>1 TFFF interrupts or DMA requests are enabled.                                              |
| 7<br>TFFF_DIRS | Transmit FIFO Fill DMA or Interrupt Request Select<br><br>Selects between generating a DMA request or an interrupt request. When SR[TFFF] and RSER[TFFF_RE] are set, this field selects between generating an interrupt request or a DMA request.<br><br>0 TFFF flag generates interrupt requests.<br>1 TFFF flag generates DMA requests. |
| 8<br>Reserved  | Always write the reset value to this field.<br><br>This field is reserved.                                                                                                                                                                                                                                                                |
| 9<br>DPEF_RE   | DSI Parity Error Request Enable<br><br>Enables the DPEF flag in the SR to generate an interrupt request.<br><br>0 DPEF interrupt requests are disabled.<br>1 DPEF interrupt requests are enabled.                                                                                                                                         |
| 10<br>SPEF_RE  | D SPI Parity Error Request Enable<br><br>Enables the SPEF flag in the SR to generate an interrupt request.<br><br>0 SPEF interrupt requests are disabled.<br>1 SPEF interrupt requests are enabled.                                                                                                                                       |
| 11<br>DDIF_RE  | DSI data received with active bits Request Enable<br><br>Enables the SR[DDIF] flag to generate an interrupt request.<br><br>0 Disabled<br>1 Enabled                                                                                                                                                                                       |
| 12<br>RFOF_RE  | Receive FIFO Overflow Request Enable<br><br>Enables the RFOF flag in the SR to generate an interrupt request.<br><br>0 RFOF interrupt requests are disabled.<br>1 RFOF interrupt requests are enabled.                                                                                                                                    |
| 13<br>Reserved | Always write the reset value to this field.<br><br>This field is reserved.                                                                                                                                                                                                                                                                |
| 14<br>RFDF_RE  | Receive FIFO Drain Request Enable                                                                                                                                                                                                                                                                                                         |

*Table continues on the next page...*

**DSPI\_RSER field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                    |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | Enables the RFDF flag in the SR to generate a request. The RFDF_DIRS bit selects between generating an interrupt request or a DMA request.<br><br>0 RFDF interrupt or DMA requests are disabled.<br>1 RFDF interrupt or DMA requests are enabled.                                                                                              |
| 15<br>RFDF_DIRS   | Receive FIFO Drain DMA or Interrupt Request Select<br><br>Selects between generating a DMA request or an interrupt request. When the RFDF flag bit in the SR is set, and the RFDF_RE bit in the RSER is set, the RFDF_DIRS bit selects between generating an interrupt request or a DMA request.<br><br>0 Interrupt request.<br>1 DMA request. |
| 16<br>Reserved    | Always write the reset value to this field.<br><br>This field is reserved.                                                                                                                                                                                                                                                                     |
| 17<br>Reserved    | Always write the reset value to this field.<br><br>This field is reserved.                                                                                                                                                                                                                                                                     |
| 18–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                        |

**39.4.7 PUSH TX FIFO Register In Master Mode (DSPI\_PUSHR)**

Specifies data to be transferred to the TX FIFO and CMD FIFO. User must write 16-bits data into TXDATA field. An 8- or 16-bit write access to the TXDATA field transfers 16 bits of data bus to the TX FIFO. A write access to the command fields transfers the 16 bits of command information to the CMD FIFO. In Master mode, the register transfers 16 bits of data to the TX FIFO and 16 bits of command information to the CMD FIFO.

The TX FIFO and CMD FIFO must be filled simultaneously. In other words, you must perform write accesses to both the data and command fields for every PUSHR operation. Because both the TX FIFO and CMD FIFO are written to and read from simultaneously, they behave as a single 32 bit FIFO.

A read access of PUSHR returns the topmost TX FIFO and CMD FIFO entries concatenated.

When the module is disabled, writing to this register does not update the FIFO. Therefore, any reads performed while the module is disabled return the last PUSHR write performed while the module was still enabled.



Address: 0h base + 34h offset = 34h

|       |      |   |   |   |   |   |   |   |          |   |     |    |    |    |    |    |
|-------|------|---|---|---|---|---|---|---|----------|---|-----|----|----|----|----|----|
| Bit   | 0    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8        | 9 | 10  | 11 | 12 | 13 | 14 | 15 |
| R     | CONT |   |   |   |   |   |   |   | Reserved |   | PCS |    |    |    |    |    |
| W     |      |   |   |   |   |   |   |   |          |   |     |    |    |    |    |    |
| Reset | 0    | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0        | 0 | 0   | 0  | 0  | 0  | 0  | 0  |

|       |        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|--------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 16     | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | TXDATA |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0      | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**DSPI\_PUSHR field descriptions**

| Field       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>CONT   | <p>Continuous Peripheral Chip Select Enable</p> <p>Selects a continuous selection format. The bit is used in DSPI Master mode. The bit enables the selected PCS signals to remain asserted between transfers.</p> <p>0 Return PCSn signals to their inactive state between transfers.<br/>1 Keep PCSn signals asserted between transfers.</p>                                                                                                                                              |
| 1–3<br>CTAS | <p>Clock and Transfer Attributes Select</p> <p>Selects which CTAR to use in master mode to specify the transfer attributes for the associated DSPI frame. In SPI Slave mode, CTAR0 is used. See the chip specific section for details to determine how many CTARs this device has. You should not program a value in this field for a register that is not present.</p> <p>000 CTAR0<br/>001 CTAR1<br/>010 CTAR2<br/>011 CTAR3<br/>100 CTAR4<br/>101 CTAR5<br/>110 CTAR6<br/>111 CTAR7</p> |
| 4<br>EOQ    | <p>End Of Queue</p> <p>Host software uses this bit to signal to the module that the current DSPI transfer is the last in a queue. At the end of the transfer, the EOQF bit in the SR is set.</p> <p>0 The DSPI data is not the last data to transfer.<br/>1 The<br/>D<br/>SPI data is the last data to transfer.</p>                                                                                                                                                                       |
| 5<br>CTCNT  | <p>Clear Transfer Counter</p> <p>Clears the TCNT field in the TCR register. The TCNT field is cleared before the module starts transmitting the current SPI frame.</p>                                                                                                                                                                                                                                                                                                                     |

*Table continues on the next page...*

**DSPI\_PUSHR field descriptions (continued)**

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                 | 0 Do not clear the TCR[TCNT] field.<br>1 Clear the TCR[TCNT] field.                                                                                                                                                                                                                                                                                                                                      |
| 6<br>PE         | Parity Enable<br><br>Enables parity bit transmission and parity reception check for the DSPI frame.<br><br>0 No parity bit included/checked.<br>1 Parity bit is transmitted instead of the last data bit in the frame; parity is checked for the received frame.                                                                                                                                         |
| 7<br>PP         | Parity Polarity<br><br>Controls the polarity of the parity bit transmitted and checked.<br><br>0 Even Parity: the number of 1 bits in the transmitted frame is even. The SR[SPEF] bit is set if the number of 1 bits is odd in the received frame.<br>1 Odd Parity: the number of 1 bits in the transmitted frame is odd. The SR[SPEF] bit is set if the number of 1 bits is even in the received frame. |
| 8–9<br>Reserved | Always write the reset value to this field.<br><br>This field is reserved.                                                                                                                                                                                                                                                                                                                               |
| 10–15<br>PCS    | Select which PCS signals are to be asserted for the transfer. Refer to the chip-specific DSPI information for the number of PCS signals used in this chip.<br><br>0 Negate the PCS[x] signal.<br>1 Assert the PCS[x] signal.                                                                                                                                                                             |
| 16–31<br>TXDATA | Transmit Data<br><br>Holds SPI data to be transferred according to the associated DSPI command.                                                                                                                                                                                                                                                                                                          |

**39.4.8 PUSH TX FIFO Register In Slave Mode (DSPI\_PUSHR\_SLAVE)**

Specifies data to be transferred to the TX FIFO in slave mode. An 8- or 16-bit write access to PUSHR transfers the 16-bit TXDATA field to the TX FIFO.

Address: 0h base + 34h offset = 34h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**DSPI\_PUSHR\_SLAVE field descriptions**

| Field            | Description                                                                             |
|------------------|-----------------------------------------------------------------------------------------|
| 0–15<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |

*Table continues on the next page...*

**DSPI\_PUSHR\_SLAVE field descriptions (continued)**

| Field           | Description                                                                                      |
|-----------------|--------------------------------------------------------------------------------------------------|
| 16–31<br>TXDATA | Transmit Data<br><br>Holds DSPI data to be transferred according to the associated DSPI command. |

**39.4.9 POP RX FIFO Register (DSPI\_POPR)**

POPR is used to read the RX FIFO. Eight- or sixteen-bit read accesses to the POPR have the same effect on the RX FIFO as 32-bit read accesses. A write to this register will generate a Transfer Error.

Address: 0h base + 38h offset = 38h

|       |        |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|--------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0      | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | RXDATA |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |        |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0      | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**DSPI\_POPR field descriptions**

| Field          | Description                                                                                                   |
|----------------|---------------------------------------------------------------------------------------------------------------|
| 0–31<br>RXDATA | Received Data<br><br>Contains the DSPI data from the RX FIFO entry to which the Pop Next Data Pointer points. |

**39.4.10 Transmit FIFO Registers (DSPI\_TXFRn)**

TXFRn registers provide visibility into the TX FIFO for debugging purposes. Each register is an entry in the TX FIFO. The registers are read-only and cannot be modified. Reading the TXFRx registers does not alter the state of the TX FIFO.

Address: 0h base + 3Ch offset + (4d × i), where i=0d to 3d

|       |              |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|--------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|--------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0            | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16     | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | TXCMD_TXDATA |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | TXDATA |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |              |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0            | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0      | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

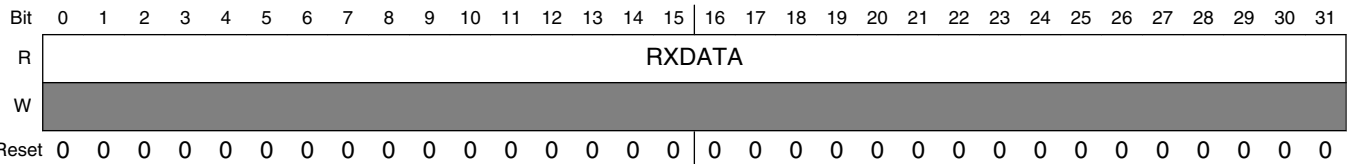
DSPI\_TXFRn field descriptions

| Field                    | Description                                                                                                                                                                                 |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>TXCMD_<br>TXDATA | Transmit Command or Transmit Data<br><br>In Master mode the TXCMD field contains the command that sets the transfer attributes for the DSPI data.<br>In Slave mode, this field is reserved. |
| 16–31<br>TXDATA          | Transmit Data<br><br>Contains the DSPI data to be shifted out.                                                                                                                              |

39.4.11 Receive FIFO Registers (DSPI\_RXFRn)

RXFRn provide visibility into the RX FIFO for debugging purposes. Each register is an entry in the RX FIFO. The RXFR registers are read-only. Reading the RXFRx registers does not alter the state of the RX FIFO.

Address: 0h base + 7Ch offset + (4d × i), where i=0d to 3d



DSPI\_RXFRn field descriptions

| Field          | Description                                          |
|----------------|------------------------------------------------------|
| 0–31<br>RXDATA | Receive Data<br><br>Contains the received DSPI data. |

### 39.4.12 DSI Configuration Register 0 (DSPI\_DSICR0)

Selects various attributes associated with the following configurations:

- DSI
- TSB

Do not write to the DSICR0 while the module is in the Running state.

Address: 0h base + BCh offset = BCh

|       |          |       |          |   |   |   |   |   |   |      |      |      |      |     |    |    |
|-------|----------|-------|----------|---|---|---|---|---|---|------|------|------|------|-----|----|----|
| Bit   | 0        | 1     | 2        | 3 | 4 | 5 | 6 | 7 | 8 | 9    | 10   | 11   | 12   | 13  | 14 | 15 |
| R     | Reserved | FMSZ4 | Reserved |   |   |   |   |   | 0 |      |      |      |      |     |    |    |
| W     |          |       |          |   |   |   |   |   |   | TSBC | TXSS | TPOL | TRRE | CID |    |    |
| Reset | 0        | 0     | 0        | 0 | 0 | 0 | 0 | 0 | 0 |      |      | 0    | 0    | 0   | 0  | 0  |

|       |       |         |    |    |    |     |     |    |    |       |    |    |    |    |    |    |
|-------|-------|---------|----|----|----|-----|-----|----|----|-------|----|----|----|----|----|----|
| Bit   | 16    | 17      | 18 | 19 | 20 | 21  | 22  | 23 | 24 | 25    | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | DCONT | DSICTAS |    |    |    | DMS | PES | PE | PP | DPCSx |    |    |    |    |    |    |
| W     |       |         |    |    |    |     |     |    |    |       |    |    |    |    |    |    |
| Reset | 0     | 0       | 0  | 0  | 0  | 0   | 0   | 0  | 0  | 0     | 0  | 0  | 0  | 0  | 0  | 0  |

**DSPI\_DSICR0 field descriptions**

| Field           | Description                                                                                                                                                                              |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>Reserved   | This field is reserved.                                                                                                                                                                  |
| 1<br>FMSZ4      | MSB of the frame size in master mode.<br><br>If the bit is set, 16 is added to the frame size, as defined by the CTARn[FMSZ] field. The CTARx register is selected by the DSICTAS field. |
| 2–7<br>Reserved | This field is reserved.                                                                                                                                                                  |
| 8<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                  |

*Table continues on the next page...*

## DSPI\_DSICR0 field descriptions (continued)

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                  |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 11<br>TSBC       | <p>Timed Serial Bus Configuration.</p> <p>Enables the Timed Serial Bus interface configuration and allows <math>t_{DT}</math> to be programmable.</p> <p>0 Timed Serial Bus Configuration disabled.<br/>1 Timed Serial Bus Configuration enabled.</p>                                                                                                                                                                        |
| 12<br>TXSS       | <p>Transmit Data Source Select.</p> <p>Selects the source of data to be serialized. The source can be either data from host software written to the DSI Alternate Serialization Data Register (ASDR0/1) or Parallel Input pin states latched into the DSI Serialization Data Register (SDR0/1).</p> <p>0 Source of serialized data is the SDR.<br/>1 Source of serialized data is the ASDR.</p>                              |
| 13<br>TPOL       | <p>Trigger Polarity</p> <p>Selects the active edge of the hardware trigger input signal (HT), initiating DSI frames transfer.</p> <p>0 Falling edge initiates a transfer.<br/>1 Rising edge initiates a transfer.</p>                                                                                                                                                                                                        |
| 14<br>TRRE       | <p>Trigger Reception Enable</p> <p>Enables the module to initiate a DSI frames transfer with an external trigger signal.</p> <p><b>NOTE:</b> When TRIG[ENABLE] is set, the TRRE bit is given by TRIG[TRRE] which overrides DSICR0[TRRE].</p> <p>0 Trigger signal reception disabled.<br/>1 Trigger signal reception enabled.</p>                                                                                             |
| 15<br>CID        | <p>Change In Data Transfer Enable</p> <p>Enables a change in serialization data to initiate a DSI frames transfer in DSI and CSI configurations. When the CID bit is set, DSI frames are initiated when the current DSI data differs from the previous DSI data shifted out.</p> <p><b>NOTE:</b> When TRIG[ENABLE] is set, the CID bit is given by TRIG[CID] which overrides DSICR0[CID].</p>                                |
| 16<br>DCONT      | <p>DSI Continuous Peripheral Chip Select Enable</p> <p>Enables the PCS signals to remain asserted between transfers. The DCONT bit affects the PCS signals only in DSI master mode. When the TSBC bit is set, DCONT has no effect.</p> <p>0 Return Peripheral Chip Select signals to their inactive state after the transfer is complete.<br/>1 Keep Peripheral Chip Select signals asserted after transfer is complete.</p> |
| 17–19<br>DSICTAS | <p>DSI Clock and Transfer Attributes Select</p> <p>Selects the CTAR register to use to provide transfer attributes for DSI frames. The DSICTAS field is used in DSI master mode. In DSI slave mode, CTAR1 is always selected.</p>                                                                                                                                                                                            |
| 20<br>DMS        | <p>Data Match Stop</p> <p>If set, stops DSI frame transmissions if DDIF flag is set in the SR register.</p> <p>0 DDIF flag does not have an effect on DSI frame transmissions.<br/>1 DDIF flag stops DSI frame transmissions.</p>                                                                                                                                                                                            |

Table continues on the next page...

**DSPI\_DSICR0 field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                                                                                                                                                      |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 21<br>PES      | Parity Error Stop<br><br>If set, stops DSI operation if the parity error happened in the received DSI frame.<br><br>0 Parity error does not stop DSI frame transmissions.<br>1 Parity error stops all DSI frame transmissions.                                                                                                                                                                   |
| 22<br>PE       | Parity Enable<br><br>Enables parity bit transmission and parity reception check for the DSI frames.<br><br>0 No parity bit included/checked.<br>1 Parity bit is transmitted instead of the last data bit in the frame; parity is checked for the received frame.                                                                                                                                 |
| 23<br>PP       | Parity Polarity<br><br>Controls the polarity of the parity bit transmitted and checked.<br><br>0 Even Parity: the number of 1 bits in the transmitted frame is even. The SR[DPEF] bit is set if in the received frame number of 1 bits is odd.<br>1 Odd Parity: the number of 1 bits in the transmitted frame is odd. The SR[DPEF] bit is set if in the received frame number of 1 bits is even. |
| 24–31<br>DPCSx | DSI Peripheral Chip Select 0–7<br><br>Selects which of the PCS signals to assert during a DSI master mode transfer.<br><br>0 Negate PCS[x].<br>1 Assert PCS[x].                                                                                                                                                                                                                                  |

**39.4.13 DSI Serialization Data Register 0 (DSPI\_SDR0)**

SDR0 contains the states of the 32 LSB Parallel Input signals. The states of the Parallel Input signals are latched into the SDR0 on the rising edge of every bus clock. The SDR0 is read-only. When the TXSS bit in the DSICR0 is cleared, the data in the SDR0 is used as the source of the DSI frames.

Address: 0h base + C0h offset = C0h

|       |          |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0        | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | SER_DATA |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |          |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**DSPI\_SDR0 field descriptions**

| Field            | Description                                                                         |
|------------------|-------------------------------------------------------------------------------------|
| 0–31<br>SER_DATA | Serialized Data<br>Contains the signal states of the 32 LSB Parallel Input signals. |

**39.4.14 DSI Alternate Serialization Data Register 0 (DSPI\_ASDR0)**

ASDR0 is used by host software to write the 32 LSB of the data to be serialized. When the TXSS bit in the DSICR0 is set, the data in the ASDR0 is the source of the DSI frames. Writes to the ASDR0 take effect on the next frame boundary.

Address: 0h base + C4h offset = C4h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**DSPI\_ASDR0 field descriptions**

| Field             | Description                                                                           |
|-------------------|---------------------------------------------------------------------------------------|
| 0–31<br>ASER_DATA | Alternate Serialized Data<br>Holds the alternate 32 LSB of the data to be serialized. |

**39.4.15 DSI Transmit Comparison Register 0 (DSPI\_COMPR0)**

COMPR0 holds a copy of the 32 LSB of the last transmitted DSI data. COMPR0 is read-only. DSI data is transferred to this register as it is loaded into the TX Shift Register.

Address: 0h base + C8h offset = C8h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**DSPI\_COMPR0 field descriptions**

| Field             | Description  |
|-------------------|--------------|
| 0–31<br>COMP_DATA | Compare Data |



**DSPI\_COMPR0 field descriptions (continued)**

| Field | Description                                       |
|-------|---------------------------------------------------|
|       | Holds the 32 LSB of the last serialized DSI data. |

**39.4.16 DSI Deserialization Data Register 0 (DSPI\_DDR0)**

DDR0 holds the 32 LSB signal states for the Parallel Output signals. DDR0 is read-only and host software can read data from incoming DSI frames.

Address: 0h base + CCh offset = CCh

|       |            |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0          | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | DESER_DATA |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |            |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0          | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**DSPI\_DDR0 field descriptions**

| Field              | Description                                                                                                               |
|--------------------|---------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>DESER_DATA | Deserialized Data<br>Holds 32 LSB of deserialized data that is presented as signal states to the Parallel Output signals. |

### 39.4.17 DSI Configuration Register 1 (DSPI\_DSICR1)

Selects various attributes associated with the following configurations:

- DSI
- TSB

The user must not write to the DSICR1 while the module is in the Running state.

Address: 0h base + D0h offset = D0h

|       |   |   |   |        |   |   |   |   |   |   |    |    |    |      |      |    |
|-------|---|---|---|--------|---|---|---|---|---|---|----|----|----|------|------|----|
| Bit   | 0 | 1 | 2 | 3      | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13   | 14   | 15 |
| R     | 0 |   |   | TSBCNT |   |   |   |   | 0 |   |    | 0  | 0  | DSE1 | DSE0 |    |
| W     |   |   |   |        |   |   |   |   |   |   |    |    |    |      |      |    |
| Reset | 0 | 0 | 0 | 0      | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  |    | 0    | 0    | 0  |

|       |        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|--------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 16     | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |        |    |    |    |    |    |    |    | 0  |    |    |    |    |    |    |    |
| W     | TRGPRD |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0      | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

#### DSPI\_DSICR1 field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                       |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–2<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                           |
| 3–7<br>TSBCNT    | Timed Serial Bus Operation Count<br><br>When TSB is enabled (DSICR0[TSBC]), TSBCNT defines the length of the data frame. The length value can vary from 3 to 31 (00011b to 11111b). The length value specifies the number of data bits to be shifted out during a transfer in TSB mode. The number of data bits in the data frame is one more than the value in the TSBCNT field. |
| 8–10<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                           |
| 11<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                           |
| 13<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                           |
| 14<br>DSE1       | Data Select Enable 1<br><br>When the TBSC bit is set, the DSE1 bit controls insertion of the zero bit (Data Select) in the middle of the data frame. The insertion bit position is defined by the FMSZ field of CTARn register, selected by the DSICR0[DSICTAS] field. The TSBCNT field value must be greater than the position of insertion for proper operation of DSE1.        |

Table continues on the next page...

**DSPI\_DSICR1 field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | 0 No Zero bit is inserted in the middle of the data frame.<br>1 Zero bit is inserted at the middle of the data frame. Total number of bits in the data frame is increased by 1.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 15<br>DSE0        | Data Select Enable 0<br><br>When the TBSC bit is set, the DSE1 bit controls insertion of the zero bit (Data Select) in the beginning of the DSI frame.<br><br>0 No Zero bit is inserted in the beginning of the frame.<br>1 Zero bit is inserted at the beginning of the data frame. Total number of bits in the data frame is increased by 1.                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 16–23<br>TRGPRD   | Internal Trigger Period for the ITSB mode.<br><br>When DSICR0[TRG] bit is cleared and ITSB Mode is enabled, this field determines the trigger period for the internal trigger in number of baud rate clock cycles. The trigger period should be calculated as:<br>Trigger Period = 32 bits (maximum DSI frame size) + 2 selection bits (maximum) + $t_{pF}$ (passive frame time)<br><br>See CTAR[FMSZ] for the minimum allowed frame size. See <a href="#">Interleaved TSB (ITSB) Mode</a> for more details.<br><br>0 - 4 These values are invalid since minimum Command frame size is 5. (ie. 1 selction bit + minimum frame size which is 4).<br>5 - 255 Number of baud rate clock cycles to be used as the trigger period. The trigger period implemented is equal to TRGPRD + 1. |
| 16–23<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 24–31<br>DPCS1_x  | DSI Peripheral Chip Select 0–7<br><br>DPCS1 selects the PCS signals to assert during the second part of the DSI frame.<br><br>0 Negate PCS[x].<br>1 Assert PCS[x].                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

**39.4.18 DSI Serialization Source Select Register 0 (DSPI\_SSR0)**

SSR0 is used to create a combined frame for transmission that contains bits from ASDR0 and SDR0. Each bit in the SSR0 register selects a corresponding bit to be serialized. When DSICR0[TXSS] is set, the SSR0 register value has no effect.

Address: 0h base + D4h offset = D4h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**DSPI\_SSR0 field descriptions**

| Field      | Description                                                                                                                                                                                                                                                                                                            |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>SS | Source Select<br><br>Select the serialization source for the 32 LSB of the DSI frame. Each SS bit selects data for a corresponded bit in the transmitted frame.<br><br>0 The bit in the transmitted frame is taken from the Parallel Input pin.<br>1 The bit in the transmitted frame is taken from the ASDR0 register |

**39.4.19 DSI Parallel Input Select Register 0 (DSPI\_PISR0)**

PISR0–3 select each data bit for the transmitted frame from 32 parallel input pins. Each Input Pin Select (IPS) field controls one bit in the transmitted frame. Each register contains control fields for 8 bits in the frame. The select field value is defined as a 4-bit signed integer number. The selected parallel input pin number is defined as a difference between the sum of the field number and the field value.

For example, if IPS16 is equal to the binary number 1111 (minus 1 decimal), then bit 16 in the frame is taken from parallel input pin number 15. When the IPS0 is equal to -1, then bit 0 in the frame is taken from parallel input 31. When the IPS0 is equal to +1, then bit 0 in the frame is taken from Parallel Input 1 and so on.

Note that PISR0–3 only preselect the Parallel Input pins. The final selection to the transmitted frame is made with the SSR0-1 register bits or the DSICR0[TXSS] bit.

Address: 0h base + D8h offset = D8h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**DSPI\_PISR0 field descriptions**

| Field         | Description                                                                           |
|---------------|---------------------------------------------------------------------------------------|
| 0–3<br>IPS7   | Input Pin Select 7<br><br>Selects the Parallel Input pin for transmitted frame bit 7. |
| 4–7<br>IPS6   | Input Pin Select 6<br><br>Selects the Parallel Input pin for transmitted frame bit 6. |
| 8–11<br>IPS5  | Input Pin Select 5<br><br>Selects the Parallel Input pin for transmitted frame bit 5. |
| 12–15<br>IPS4 | Input Pin Select 5                                                                    |

*Table continues on the next page...*

**DSPI\_PISR0 field descriptions (continued)**

| Field         | Description                                                                       |
|---------------|-----------------------------------------------------------------------------------|
|               | Selects the Parallel Input pin for transmitted frame bit 4.                       |
| 16–19<br>IPS3 | Input Pin Select 3<br>Selects the Parallel Input pin for transmitted frame bit 3. |
| 20–23<br>IPS2 | Input Pin Select 2<br>Selects the Parallel Input pin for transmitted frame bit 2. |
| 24–27<br>IPS1 | Input Pin Select 1<br>Selects the Parallel Input pin for transmitted frame bit 1. |
| 28–31<br>IPS0 | Input Pin Select 0<br>Selects the Parallel Input pin for transmitted frame bit 0. |

**39.4.20 DSI Parallel Input Select Register 1 (DSPI\_PISR1)**

Address: 0h base + DCh offset = DCh

|       |       |   |   |   |       |   |   |   |       |   |    |    |       |    |    |    |       |    |    |    |       |    |    |    |      |    |    |    |      |    |    |    |
|-------|-------|---|---|---|-------|---|---|---|-------|---|----|----|-------|----|----|----|-------|----|----|----|-------|----|----|----|------|----|----|----|------|----|----|----|
| Bit   | 0     | 1 | 2 | 3 | 4     | 5 | 6 | 7 | 8     | 9 | 10 | 11 | 12    | 13 | 14 | 15 | 16    | 17 | 18 | 19 | 20    | 21 | 22 | 23 | 24   | 25 | 26 | 27 | 28   | 29 | 30 | 31 |
| R     | IPS15 |   |   |   | IPS14 |   |   |   | IPS13 |   |    |    | IPS12 |    |    |    | IPS11 |    |    |    | IPS10 |    |    |    | IPS9 |    |    |    | IPS8 |    |    |    |
| W     |       |   |   |   |       |   |   |   |       |   |    |    |       |    |    |    |       |    |    |    |       |    |    |    |      |    |    |    |      |    |    |    |
| Reset | 0     | 0 | 0 | 0 | 0     | 0 | 0 | 0 | 0     | 0 | 0  | 0  | 0     | 0  | 0  | 0  | 0     | 0  | 0  | 0  | 0     | 0  | 0  | 0  | 0    | 0  | 0  | 0  | 0    | 0  | 0  | 0  |

**DSPI\_PISR1 field descriptions**

| Field          | Description                                                                         |
|----------------|-------------------------------------------------------------------------------------|
| 0–3<br>IPS15   | Input Pin Select 15<br>Selects the Parallel Input pin for transmitted frame bit 15. |
| 4–7<br>IPS14   | Input Pin Select 14<br>Selects the Parallel Input pin for transmitted frame bit 14. |
| 8–11<br>IPS13  | Input Pin Select 13<br>Selects the Parallel Input pin for transmitted frame bit 13. |
| 12–15<br>IPS12 | Input Pin Select 12<br>Selects the Parallel Input pin for transmitted frame bit 12. |
| 16–19<br>IPS11 | Input Pin Select 11<br>Selects the Parallel Input pin for transmitted frame bit 11. |
| 20–23<br>IPS10 | Input Pin Select 10<br>Selects the Parallel Input pin for transmitted frame bit 10. |
| 24–27<br>IPS9  | Input Pin Select 9<br>Selects the Parallel Input pin for transmitted frame bit 9.   |

*Table continues on the next page...*

**DSPI\_PISR1 field descriptions (continued)**

| Field         | Description                                                                       |
|---------------|-----------------------------------------------------------------------------------|
| 28–31<br>IPS8 | Input Pin Select 8<br>Selects the Parallel Input pin for transmitted frame bit 8. |

**39.4.21 DSI Parallel Input Select Register 2 (DSPI\_PISR2)**

Address: 0h base + E0h offset = E0h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**DSPI\_PISR2 field descriptions**

| Field          | Description                                                                         |
|----------------|-------------------------------------------------------------------------------------|
| 0–3<br>IPS23   | Input Pin Select 23<br>Selects the Parallel Input pin for transmitted frame bit 23. |
| 4–7<br>IPS22   | Input Pin Select 22<br>Selects the Parallel Input pin for transmitted frame bit 22. |
| 8–11<br>IPS21  | Input Pin Select 21<br>Selects the Parallel Input pin for transmitted frame bit 21. |
| 12–15<br>IPS20 | Input Pin Select 20<br>Selects the Parallel Input pin for transmitted frame bit 20. |
| 16–19<br>IPS19 | Input Pin Select 19<br>Selects the Parallel Input pin for transmitted frame bit 19. |
| 20–23<br>IPS18 | Input Pin Select 18<br>Selects the Parallel Input pin for transmitted frame bit 18. |
| 24–27<br>IPS17 | Input Pin Select 17<br>Selects the Parallel Input pin for transmitted frame bit 17. |
| 28–31<br>IPS16 | Input Pin Select 16<br>Selects the Parallel Input pin for transmitted frame bit 16. |

### 39.4.22 DSI Parallel Input Select Register 3 (DSPI\_PISR3)

Address: 0h base + E4h offset = E4h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

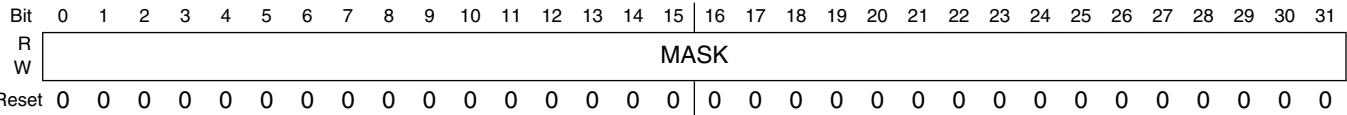
#### DSPI\_PISR3 field descriptions

| Field          | Description                                                                         |
|----------------|-------------------------------------------------------------------------------------|
| 0–3<br>IPS31   | Input Pin Select 31<br>Selects the Parallel Input pin for transmitted frame bit 31. |
| 4–7<br>IPS30   | Input Pin Select 30<br>Selects the Parallel Input pin for transmitted frame bit 30. |
| 8–11<br>IPS29  | Input Pin Select 29<br>Selects the Parallel Input pin for transmitted frame bit 29. |
| 12–15<br>IPS28 | Input Pin Select 28<br>Selects the Parallel Input pin for transmitted frame bit 28. |
| 16–19<br>IPS27 | Input Pin Select 27<br>Selects the Parallel Input pin for transmitted frame bit 27. |
| 20–23<br>IPS26 | Input Pin Select 26<br>Selects the Parallel Input pin for transmitted frame bit 26. |
| 24–27<br>IPS25 | Input Pin Select 25<br>Selects the Parallel Input pin for transmitted frame bit 25. |
| 28–31<br>IPS24 | Input Pin Select 24<br>Selects the Parallel Input pin for transmitted frame bit 24. |

39.4.23 DSI Deserialized Data Interrupt Mask Register 0 (DSPI\_DIMR0)

DIMR0 selects bits in the 32 least-significant bits of the received DSI frame to be checked to generate the DDI interrupt.

Address: 0h base + E8h offset = E8h



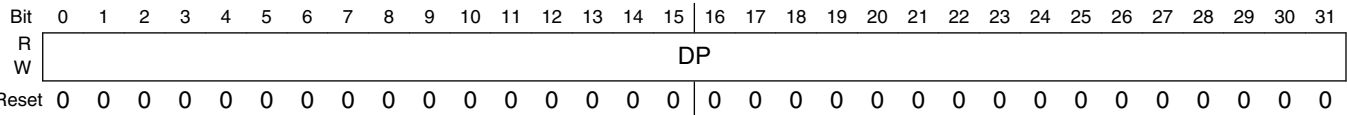
DSPI\_DIMR0 field descriptions

| Field        | Description                                                                                                                                                                                                                                                                                                                                        |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>MASK | Mask<br><br>Defines which bits in the 32 LSB of received deserialization data should be checked to produce the Deserialized Data Interrupt (DDI).<br><br>0 The bit in the received DSI frame does not produce a DDI interrupt.<br>1 The bit in the received DSI frame can produce a DDI interrupt if the data bit matches the configured polarity. |

39.4.24 DSI Deserialized Data Polarity Interrupt Register 0 (DSPI\_DPIR0)

DPIR0 defines which data bit value in the 32 least-significant bits of the received DSI frame generates the DDI interrupt.

Address: 0h base + ECh offset = ECh



DSPI\_DPIR0 field descriptions

| Field      | Description                                                                                                    |
|------------|----------------------------------------------------------------------------------------------------------------|
| 0–31<br>DP | Data Polarity<br><br>Defines which value of the 32 LSB of received deserialization data sets the SR[DDIF] bit. |



**DSPI\_DPIR0 field descriptions (continued)**

| Field | Description                                        |
|-------|----------------------------------------------------|
| 0     | If the received bit is 0, the SR[DDIF] bit is set. |
| 1     | If the received bit is 1, the SR[DDIF] bit is set. |

## 39.5 Functional description

The module supports full-duplex, synchronous serial communications between chips and peripheral devices. The module can also be used to reduce the number of pins required for I/O by serializing and deserializing Parallel Input/Output signals. All communications are done with SPI-like protocol.

The module has the following configurations

- The DSPI Configuration in which the module operates as a basic DSPI or a queued SPI.
- DSI Configuration in which the module serializes and deserializes Parallel Input/Output signals or bits from memory-mapped registers
- When Master mode is enabled, the module also supports CSI Configuration, which combines the functionality of the DSPI and DSI configurations.

The DCONF field in the Module Configuration Register (MCR) determines the module Configuration. See [Module Configuration Register \(DSPI\\_MCR\)](#) for the DSPI configuration values.

The CTAR<sub>n</sub> registers hold clock and transfer attributes. The DSPI configuration allows to select which CTAR to use on a frame by frame basis by setting a field in the DSPI command.

The DSI configuration statically selects which CTAR to use. In CSI, Configuration priority logic determines if DSPI data or DSI data is transferred and dictates what CTAR register is used for the data transfer. See [Clock and Transfer Attributes Register \(In Master Mode\) \(DSPI\\_CTAR<sub>n</sub>\)](#) for information on the fields of CTAR registers.

Typical master to slave connections are shown in the following figure. When a data transfer operation is performed, data is serially shifted a predetermined number of bit positions. Because the modules are linked, data is exchanged between the master and the slave. The data that was in the master shift register is now in the shift register of the slave, and vice versa. At the end of a transfer, the Transfer Control Flag(TCF) bit in the Shift Register(SR) is set to indicate a completed frame transfer.

**NOTE**

SPI is used for both SPI Mode and DSI mode in this diagram.

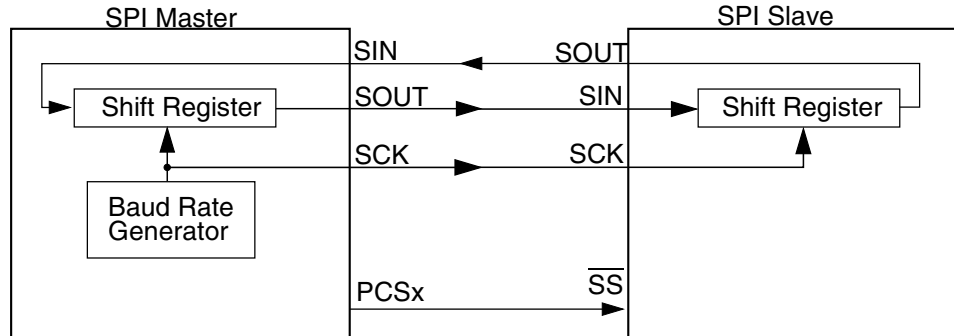


**Figure 39-45. Serial protocol overview**

Generally, more than one slave device can be connected to the module master. 6 Peripheral Chip Select (PCS) signals of the module masters can be used to select which of the slaves to communicate with. Refer to the chip specific section for details on the number of PCS signals used in this chip.

The three DSPI configurations share transfer protocol and timing properties which are described independently of the configuration in [Transfer formats](#). The transfer rate and delay settings are described in [Module baud rate and clock delay generation](#).

### 39.5.1 Start and Stop of module transfers

The module has two operating states: Stopped and Running. Both the states are independent of it's configuration. The default state of the module is Stopped. In the Stopped state, no serial transfers are initiated in Master mode and no transfers are responded to in Slave mode. The Stopped state is also a safe state for writing the various configuration registers of the module without causing undetermined results. In the Running state serial transfers take place.

The TXRXS bit in the SR indicates the state of module. The bit is set if the module is in Running state.

The module starts or transitions to Running when all of the following conditions are true:

- SR[EOQF] bit is clear
- Chip is not in the Debug mode or the MCR[FRZ] bit is clear
- MCR[HALT] bit is clear

The module stops or transitions from Running to Stopped after the current frame when any one of the following conditions exist:

- SR[EOQF] bit is set
- Chip in the Debug mode and the MCR[FRZ] bit is set
- MCR[HALT] bit is set

State transitions from Running to Stopped occur on the next frame boundary if a transfer is in progress, or immediately if no transfers are in progress.

## 39.5.2 Serial Peripheral Interface DSPI configuration

The DSPI configuration transfers data serially using a shift register and a selection of programmable transfer attributes. The module is in DSPI configuration when the DCONF field in the MCR is 0b00. The DSPI frames can be 32 bits long. The host CPU or a DMA controller transfers the DSPI data from the external to the module RAM queues to a TX FIFO buffer. The received data is stored in entries in the RX FIFO buffer. The host CPU or the DMA controller transfers the received data from the RX FIFO to memory external to the module. The operation of FIFO buffers is described in the following sections:

- [Transmit First In First Out \(TX FIFO\) buffering mechanism](#)
- [Command First In First Out \(CMD FIFO\) Buffering Mechanism](#)
- [Receive First In First Out \(RX FIFO\) buffering mechanism](#)

The interrupt and DMA request conditions are described in [Interrupts/DMA requests](#).

The DSPI configuration supports two block-specific modes—Master mode and Slave mode. In Master mode the module initiates and controls the transfer according to the fields of the executing DSPI Command. In Slave mode, the module responds only to transfers initiated by a bus master external to it and the DSPI command field space is reserved.

### 39.5.2.1 Master mode

In DSPI Master mode, the module initiates the serial transfers by controlling the SCK and the PCS signals. The executing DSPI Command determines which CTARs will be used to set the transfer attributes and which PCS signals to assert. The command field also contains various bits that help with queue management and transfer protocol. See [PUSH TX FIFO Register In Master Mode \(DSPI\\_PUSHR\)](#) for details on the DSPI command fields. The data in the executing TX FIFO entry is loaded into the shift register and

shifted out on the Serial Out (SOUT) pin. In DSPI Master mode, each DSPI frame to be transmitted has a command associated with it, allowing for transfer attribute control on a frame by frame basis.

### 39.5.2.2 Slave mode

In DSPI Slave mode the module responds to transfers initiated by an DSPI bus master. It does not initiate transfers. Certain transfer attributes such as clock polarity, clock phase, and frame size must be set for successful communication with an DSPI master. The DSPI Slave mode transfer attributes are set in the CTAR0. The data is shifted out with MSB first. Shifting out of LSB is not supported in this mode.

### 39.5.2.3 FIFO disable operation

The FIFO disable mechanisms allow DSPI transfers without using the TX FIFO, CMD FIFO or RX FIFO. The module operates as a double-buffered simplified DSPI when the FIFOs are disabled. The Transmit and Receive side of the FIFOs are disabled separately. Setting the MCR[DIS\_TXF] bit disables the TX FIFO and CMD FIFO, and setting the MCR[DIS\_RXF] bit disables the RX FIFO.

The FIFO disable mechanisms are transparent to the user and to host software. Transmit data and commands are written to the PUSHHR and received data is read from the POPR.

When the TX FIFO and CMD FIFO are disabled:

- SR[TFFF], SR[TFUF] and SR[TXCTR] behave as if there is a one-entry FIFO
- The contents of TXFRs, SR[TXNXTPTR] are undefined

Similarly, when the RX FIFO is disabled, the RFDF, RFOF, and RXCTR fields in the SR behave as if there is a one-entry FIFO, but the contents of the RXFR registers and POPNXTPTR are undefined.

### 39.5.2.4 Transmit First In First Out (TX FIFO) buffering mechanism

The TX FIFO functions as a buffer of DSPI data for transmission. The TX FIFO holds 4 words, each consisting of DSPI data. The number of entries in the TX FIFO is device-specific. DSPI data is added to the TX FIFO by writing to the Data Field of module PUSH FIFO Register (PUSHHR). TX FIFO entries can only be removed from the TX FIFO by being shifted out or by flushing the TX FIFO.

The TX FIFO Counter field (TXCTR) in the module Status Register (SR) indicates the number of valid entries in the TX FIFO. The TXCTR is updated every time a 8- or 16-bit write takes place to PUSHHR[TXDATA] or DSPI data is transferred into the shift register from the TX FIFO.

The TXNXTPTR field indicates the TX FIFO Entry that will be transmitted during the next transfer. The TXNXTPTR field is incremented every time DSPI data is transferred from the TX FIFO to the shift register. The maximum value of the field is equal to the maximum implemented TXFR number and it rolls over after reaching the maximum.

#### 39.5.2.4.1 Filling the TX FIFO

Host software or other intelligent blocks can add (push) entries to the TX FIFO and CMD FIFO by writing to the PUSHHR. When the TX FIFO is not full, the TX FIFO Fill Flag (TFFF) in the SR is set. The TFFF bit is cleared when TX FIFO is full and the DMA controller indicates that a write to PUSHHR is complete. Writing a '1' to the TFFF bit also clears it. The TFFF can generate a DMA request or an interrupt request. See [Transmit FIFO Fill Interrupt or DMA Request](#) for details.

The module ignores attempts to push data to a full TX FIFO, and the state of the TX FIFO does not change and no error condition is indicated.

#### 39.5.2.4.2 Draining the TX FIFO

The TX FIFO entries are removed (drained) by shifting DSPI data out through the shift register. Entries are transferred from the TX FIFO to the shift register and shifted out as long as there are valid entries in the TX FIFO. Every time an entry is transferred from the TX FIFO to the shift register, the TX FIFO Counter decrements by one. At the end of a transfer, the TCF bit in the SR is set to indicate the completion of a transfer. The TX FIFO is flushed by writing a '1' to the CLR\_TXF bit in MCR.

If an external bus master initiates a transfer with a module slave while the slave's TX FIFO is empty, the Transmit FIFO Underflow Flag (TFUF) in the slave's SR is set. See [Transmit FIFO Underflow Interrupt Request](#) for details.

#### 39.5.2.5 Command First In First Out (CMD FIFO) Buffering Mechanism

The CMD FIFO functions as a buffer of DSPI command used for DSPI data transmission. The TX FIFO and CMD FIFO must be filled together, i.e. write enables should be given for both the Data and Command fields while performing a PUSHHR operation.

The CMD FIFO holds 4 words, each representing DSPI command fields. The number of entries in the CMD FIFO is device-specific. DSPI Command is added to the CMD FIFO by writing to the command field of DSPI PUSH FIFO Register (PUSHR). CMD FIFO entries can only be removed from the CMD FIFO by being shifted out (to help transmit DSPI data) or by flushing the CMD FIFO.

Every CMD FIFO entry has a corresponding single TX FIFO entry attached to it because both these FIFO's are filled simultaneously.

### **39.5.2.6 Receive First In First Out (RX FIFO) buffering mechanism**

The RX FIFO functions as a buffer for data received on the SIN pin. The RX FIFO holds 4 received DSPI data frames. The number of entries in the RX FIFO is device-specific. DSPI data is added to the RX FIFO at the completion of a transfer when the received data in the shift register is transferred into the RX FIFO. DSPI data are removed (popped) from the RX FIFO by reading the module POP RX FIFO Register (POPR). RX FIFO entries can only be removed from the RX FIFO by reading the POPR or by flushing the RX FIFO.

The RX FIFO Counter field (RXCTR) in the module's Status Register (SR) indicates the number of valid entries in the RX FIFO. The RXCTR is updated every time the POPR is read or DSPI data is copied from the shift register to the RX FIFO.

The POPNXTPTR field in the SR points to the RX FIFO entry that is returned when the POPR is read. The POPNXTPTR contains the positive offset from RXFR0 in a number of 32-bit registers. For example, POPNXTPTR equal to two means that the RXFR2 contains the received DSPI data that will be returned when the POPR is read. The POPNXTPTR field is incremented every time the POPR is read. The maximum value of the field is equal to the maximum implemented RXFR number and it rolls over after reaching the maximum.

#### **39.5.2.6.1 Filling the RX FIFO**

The RX FIFO is filled with the received DSPI data from the shift register. While the RX FIFO is not full, DSPI frames from the shift register are transferred to the RX FIFO. Every time an DSPI frame is transferred to the RX FIFO, the RX FIFO Counter is incremented by one.

If the RX FIFO and shift register are full and a transfer is initiated, the RFOF bit in the SR is set indicating an overflow condition. Depending on the state of the ROOE bit in the MCR, the data from the transfer that generated the overflow is either ignored or shifted in to the shift register. If the ROOE bit is set, the incoming data is shifted in to the shift register. If the ROOE bit is cleared, the incoming data is ignored.

### 39.5.2.6.2 Draining the RX FIFO

Host CPU or a DMA can remove (pop) entries from the RX FIFO by reading the module POP RX FIFO Register (POPR). A read of the POPR decrements the RX FIFO Counter by one. Attempts to pop data from an empty RX FIFO are ignored and the RX FIFO Counter remains unchanged. The data, read from the empty RX FIFO, is undetermined.

When the RX FIFO is not empty, the RX FIFO Drain Flag (RFDF) in the SR is set. The RFDF bit is cleared when the RX\_FIFO is empty and the DMA controller indicates that a read from POPR is complete or by writing a 1 to it.

## 39.5.3 Deserial Serial Interface (DSI) Configuration

The DSI Configuration supports pin count reduction by serializing Parallel Input signals or register bits and shifting them out in a DSPI-like protocol. The timing and transfer protocol is described in [Transfer formats](#). The received serial frames are converted to a parallel form (deserialized) and placed on the Parallel Output signals or in the DSPI\_DDR register. The various features of the DSI Configuration are set in DSPI DSI Configuration Registers (DSPI\_DSICR0/1).

The DSI frames can be from 4 to 32 bits.

### 39.5.3.1 DSI Master Mode

In DSI master mode the DSPI/DSPI initiates and controls the DSI transfers. The DSI master has different conditions that can initiate a transfer:

- Continuous
- Change in data
- Trigger signal
- Trigger signal combined with a change in data

These transfer initiation conditions are described in [DSI Transfer Initiation Control](#) . Transfer attributes are set during initialization. The DSICTAS field in the DSPI\_DSICR0 determines which of the DSPI\_CTAR registers will control the transfer attributes.

### 39.5.3.2 Slave Mode

In DSI slave mode the DSPI responds to transfers initiated by a DSPI or DSI bus master. In this mode the DSPI does not initiate DSI transfers. Certain transfer attributes such as clock polarity and phase must be set for successful communication with a DSI master. The DSI slave mode transfer attributes are set in the DSPI\_CTAR1. The data is shifted out with MSB first.

### 39.5.3.3 DSI Serialization

In the DSI Configuration, up to 32 bits can be serialized using two different sources. The TXSS bit in the DSPI\_DSICR0 selects between the DSPI DSI Serialization Data Register (DSPI\_SDR0) and the DSPI DSI Alternate Serialization Data Register (DSPI\_ASDR0) as the source of the serialized data. The DSPI\_SDR0 holds the latest 32 Parallel Input signal values which is sampled at every rising edge of the bus clock. The DSPI\_ASDR0 register is written by host software and used as an alternate source of serialized data.

The DSPI has to sample the inputs (from SDR0 or ASDR0) before it transmits a DSI frame. Due to the asynchronous clocking structure within DSPI, the sampled data (from SDR0 or ASDR0) must be synchronized into the Protocol Domain before being loaded into the Shift Register for transmission.

To prevent Clock Domain Crossing issues, inputs from SDR0 or ASDR0 should not change while they are being sampled into the Protocol Domain. This is ensured in the following manner for the ASDR0 inputs. The initial write performed on ASDR0 register will cause the 32-bit ASDR0 input to be latched into a temporary register which is then locked (further latching is prevented) until this data gets synchronized into the Protocol Domain. Once synchronized, the lock is removed and the next ASDR0 write will again trigger this process. If there was a write performed on ASDR0 while the lock was asserted, this will be taken into consideration as soon as the previous synchronization process completes and lock gets de-asserted. In case, multiple writes were performed on ASDR0 when lock was asserted, only the latest value will be considered for synchronization once the lock gets de-asserted.



The entire process explained above is also true for the SDR0 input, the only difference being that SDR0 inputs are delayed internally by a single Bus clock (register interface clock) to detect a change in the parallel input pins.

Once the synchronization process for a particular De-serial input begins, it takes up to 5 Bus clock (register interface clock) periods and 5 Protocol clock periods before the synchronization process for the next De-serial input can begin. Hence the writes performed on ASDR0 and SDR0 are constrained with this timing.

The DSPI\_PISR0-3 registers allow to change relative position of the Parallel input pins in the transmitted frame. Each transmitted frame bit can be selected from 16 adjacent Parallel Inputs by writing IPSn fields. The IPSn field is treated as 4 bit integer number, representing numbers from -8 to 7. The Parallel Input pin number, selected by IPSn field is defined by the difference between the IPSn field number (n) and the IPSn field value. If the operation result is negative the number 32 should be added. If the result is higher than 32, 32 should be subtracted from the result.

For example, IPS0, set to minus 1 (binary 1111), preselects Parallel Input 1 to 0 position in the transmitted frame.

IPS6, set to 3 (binary 0011), preselects Parallel Input 3 to be bit number 6 in the transmitted frame. The value minus 2 (1110) preselects Parallel Input 8.

IPS31, set to minus 8 (binary 1000), preselects Parallel Input 7 to be bit number 31 in the transmitted frame.

### Note

The Parallel Input pin state, to be transmitted, should be selected by TXSS or DSPI\_SSR0 register and the frame size should be higher than the bit position in the preselected frame.

DSPI\_SSR0 register provides additional way to create the frame for transmission. Each bit from this register is OR'd with the TXSS bit and controls individual transmitted bit source. This way, the transmitted frame can have any combination of the DSPI\_SDR0 and DSPI\_ASDR0 bits. This feature allows control DSPI based devices, requiring control and data fields in the frame. Control field may come from DSPI\_ASDR0 register, set by the device's CPU, while data field can be generated by device peripheral modules, like PWM timers.

A copy of the last DSI frame shifted out of the Shift Register is stored in the DSPI DSI Transmit Comparison Register (DSPI\_COMPR0). This register provides added visibility for debugging and it serves as a reference for transfer initiation control. The following figure shows the DSI Serialization logic.

**NOTE**

SPI is used for both SPI Mode and DSI mode in the following diagrams.

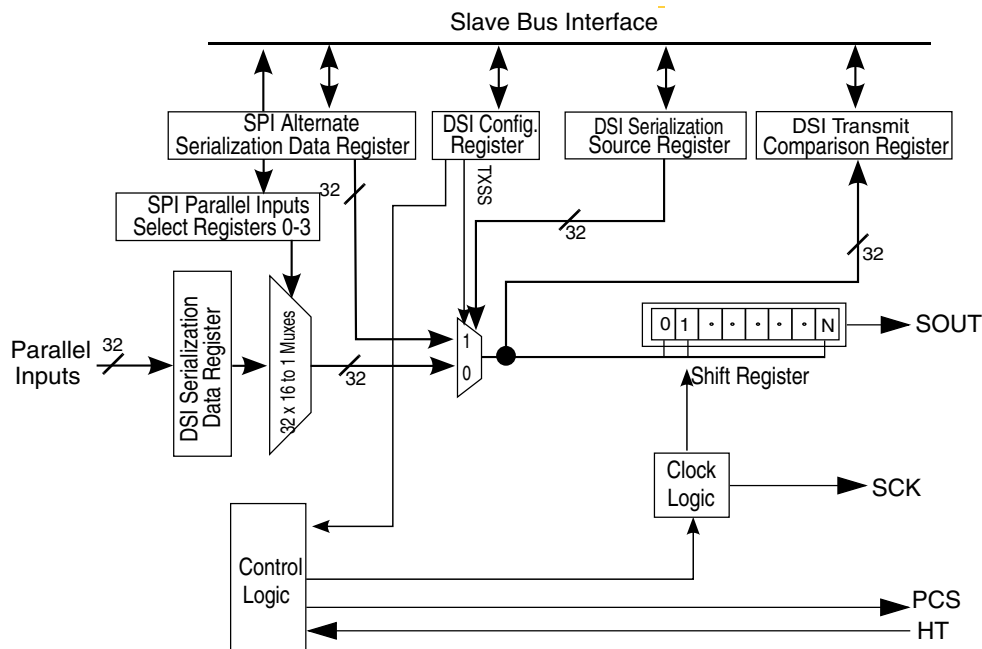


**Figure 39-46. DSI Serialization Diagram**

### 39.5.3.4 DSI Deserialization

When all bits in a DSI frame have been shifted in, the frame is copied to the DSPI DSI Deserialization Data Register (DSPI\_DDR0). This register presents the deserialized data as Parallel Output signal values. DSPI\_DDR0 is memory mapped to allow host software to read the deserialized data directly.

The received data is bit-wise compared to the value of the DSI Deserialized Data Polarity Interrupt Register (DSPI\_DPIR0), bit-wise AND'ed with DSI Deserialized Interrupt Mask Register (DSPI\_DIMR0) and the results OR'ed to produce DDIF flag in the DSPI\_SR register; which in turn can cause DDI interrupt request or DMA request based on RSER[DDIF\_RE] and RSER[DDIF\_DIRS] bits and/or stop DSI frame transmissions if DSICR0[DMS] bit is set.

The following figure shows the DSI Deserialization logic.

**NOTE**

SPI is used for both SPI Mode and DSI mode in this diagram.

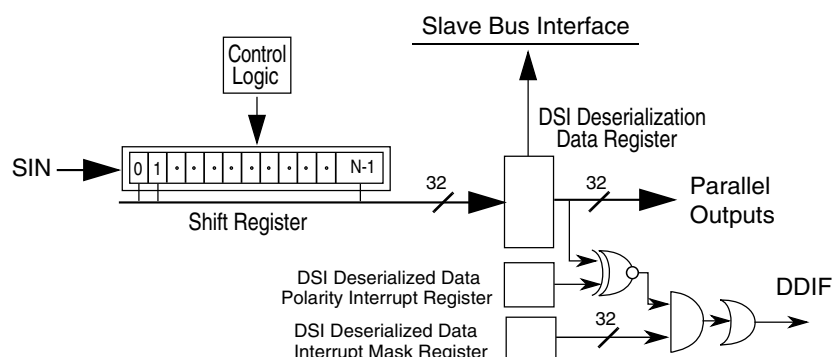


Figure 39-47. DSI Deserialization Diagram

### 39.5.3.5 DSI Transfer Initiation Control

Data transfers for a D SPI in DSI configuration are initiated by a condition. The transfer initiation conditions are selected by the TRRE and CID bits in the DSPI\_DSICR0. The following table lists the four transfer initiation conditions.

Table 39-75. DSI Data Transfer Initiation Control

| D SPI _DSICR0 Bits |     | Transfer Initiation Control |
|--------------------|-----|-----------------------------|
| TRRE               | CID |                             |
| 0                  | 0   | Continuous                  |
| 0                  | 1   | Change in Data              |
| 1                  | 0   | Triggered                   |
| 1                  | 1   | Triggered or Change in Data |

#### 39.5.3.5.1 Continuous Control

For Continuous Control a new DSI frame shifts out when the previous transfer cycle has completed and the Delay after Transfer ( $t_{DT}$ ) has elapsed. This has no effect when DSPI is in Slave Mode.

#### 39.5.3.5.2 Change In Data Control

For change in data control a transfer is initiated when the data to be serialized has changed since the transfer of the last DSI frame. A copy of the previously transferred DSI data is stored in DSPI\_COMPR0. When the data selected for the transfer from the DSPI\_SDR0 and DSPI\_AS DR0 registers is different from the data in DSPI\_COMPR0 a new DSI frame is transmitted.

## NOTE

It is not advisable to configure a DSPI slave with the combination DSICR0[DCONT] = 1, DSICR0[CID] = 1 ; since the following statement then holds true.

Once slave select is asserted, there is a very small window (of up to t module clocks) after every SOF (Start of Frame) during which if a write on ASDR/SDR registers occur, the next frame to be transmitted would be loaded into the shift register. However, it is not feasible that such a write be initiated and completed within this short span of time; hence the data transmitted by DSPIslave would lag by one data frame.

### 39.5.3.5.3 Triggered Control

For Triggered Control initiation of a transfer is controlled by the Hardware Trigger signal (HT). The TPOL bit in the DSPI\_DSICR0 selects the active edge of HT. For HT to have any affect, the TRRE bit in the DSPI\_DSICR0 must be set.

### 39.5.3.5.4 Triggered or Change In Data Control

For Triggered or Change in Data Control, initiation of a transfer is controlled by the HT signal or by the detection of a change in data to be serialized.

## 39.5.4 Combined Serial Interface (CSI) Configuration

The CSI Configuration of the DSPI is used to support DSPISPI and DSI functions on a frame by frame basis. CSI Configuration allows interleaving of DSI data frames from the parallel input signals with SPI data frames from the TX FIFO. The data returned from the bus slave is either used to drive the Parallel Output signals or it is stored in the RX FIFO. The CSI Configuration allows serialized data and configuration or diagnostic data to be transferred to a slave device using only one serial link. The DSPI is in CSI Configuration when the DCONF field in the DSPI\_MCR is 0b10. The following figure shows an example of how a DSPI can be used with a deserializing peripheral that supports SPI control for control and diagnostic frames.

## NOTE

SPI is used for both DSPISPI Mode and DSI mode in this diagram.

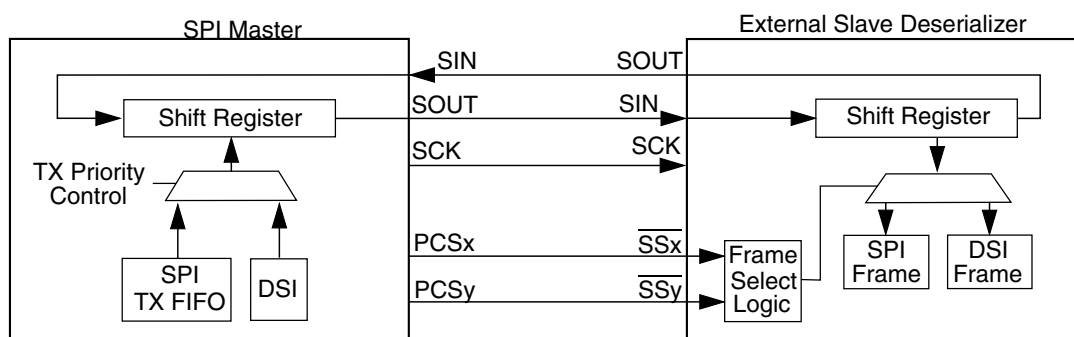


**Figure 39-48. Example of System using DSPI in CSI Configuration**

In CSI Configuration the DSPI transfers DSI data based on [DSI Transfer Initiation Control](#).

When SPI commands are in the TX FIFO, the SPI frames have priority over the DSI frames. When the TX FIFO or CMD FIFO is empty, DSI transfer resumes.

Two peripheral chip select signals indicate whether DSI data or DSPISPI data is transmitted. The user must configure the DSPI so that the two CTAR registers associated with DSI data and DSPISPI data assert different peripheral chip select signals denoted in the figure as PCSx and PCSy. The CSI Configuration is only supported in master mode.

Data returned from the external slave while a DSI frame is transferred is placed on the Parallel Output signals. Data returned from the external slave while a DSPISPI frame is transferred is moved to the RX FIFO. The TX FIFO, CMD FIFO and RX FIFO are fully functional in CSI mode.

### 39.5.4.1 CSI Serialization

Serialization in the CSI configuration is similar to serialization in DSI Configuration. The transfer attributes for DSPI frames are determined by the DSPI\_CTAR register selected by the CTAS field in the DSPI command halfword. The transfer attributes for the DSI frames are determined by the DSPI\_CTAR register selected by the DSICTAS field in the DSPI\_DSICR0.

The Parallel Inputs signal states are latched into the DSPI DSI Serialization Data Register (DSPI\_SDR0) and serialized based on the transfer initiation control settings in the DSPI\_DSICR0. When SPI frames are written to the TX FIFO (and when CMD FIFO is not empty) they have priority over DSI data from the DSPI\_SDR and are transferred at the next frame boundary. A copy of the most recently transferred DSI frame is stored in DSPI\_COMPR0. The Transfer Priority Logic selects the source of the serialized data and asserts the appropriate PCS signal.

Separate frame completion interrupts are available to indicate frame transmission completion from DSPI and DSI frames.

39.5.4.2 CSI Deserialization

The deserialized frames in CSI Configuration go into the DSPI\_DDR0 or the RX FIFO based on the transfer priority logic. When DSI frames are transferred the returned frames are deserialized and latched into the DSPI\_DDR0. When DSPI frames are transferred the returned frames are deserialized and written to the RX FIFO.

39.5.5 Module baud rate and clock delay generation

The SCK frequency and the delay values for serial transfer are generated by dividing the protocol clock frequency by a prescaler and a scaler with the option for doubling the baud rate. The following figure shows conceptually how the SCK signal is generated.

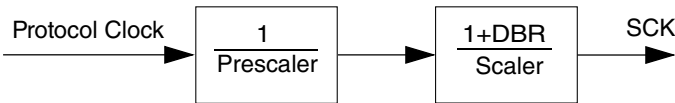


Figure 39-49. Communications clock prescalers and scalers

39.5.5.1 Baud rate generator

The baud rate is the frequency of the SCK. The protocol clock is divided by a prescaler (PBR) and scaler (BR) to produce SCK with the possibility of halving the scaler division. The DBR, PBR, and BR fields in the CTARs select the frequency of SCK by the formula in the BR field description. The following table shows an example of how to compute the baud rate.

Table 39-76. Baud rate computation example

| f <sub>p</sub> | PBR  | Prescaler | BR     | Scaler | DBR | Baud rate |
|----------------|------|-----------|--------|--------|-----|-----------|
| 100 MHz        | 0b00 | 2         | 0b0000 | 2      | 0   | 25 Mb/s   |
| 20 MHz         | 0b00 | 2         | 0b0000 | 2      | 1   | 10 Mb/s   |

NOTE

The clock frequencies mentioned in the preceding table are given as an example. Refer to the clocking chapter for the frequency used to drive this module in the device.

### 39.5.5.2 PCS to SCK Delay ( $t_{csc}$ )

The PCS to SCK delay is the length of time from assertion of the PCS signal to the first SCK edge. See [Figure 39-51](#) for an illustration of the PCS to SCK delay. The PCSSCK and CSSCK fields in the CTAR<sub>x</sub> registers select the PCS to SCK delay by the formula in the CSSCK field description. The following table shows an example of how to compute the PCS to SCK delay.

**Table 39-77. PCS to SCK delay computation example**

| $f_{sys}$ | PCSSCK | Prescaler | CSSCK  | Scaler | PCS to SCK Delay |
|-----------|--------|-----------|--------|--------|------------------|
| 100 MHz   | 0b01   | 3         | 0b0100 | 32     | 0.96 $\mu$ s     |

PCSSCK and CSSCK fields have no effect in TSB configuration.

#### NOTE

The clock frequency mentioned in the preceding table is given as an example. Refer to the clocking chapter for the frequency used to drive this module in the device.

### 39.5.5.3 After SCK Delay ( $t_{asc}$ )

The After SCK Delay is the length of time between the last edge of SCK and the negation of PCS. See [Figure 39-51](#) and [Figure 39-52](#) for illustrations of the After SCK delay. The PASC and ASC fields in the CTAR<sub>x</sub> registers select the After SCK Delay by the formula in the ASC field description. The following table shows an example of how to compute the After SCK delay.

**Table 39-78. After SCK Delay computation example**

| $f_p$   | PASC | Prescaler | ASC    | Scaler | After SCK Delay |
|---------|------|-----------|--------|--------|-----------------|
| 100 MHz | 0b01 | 3         | 0b0100 | 32     | 0.96 $\mu$ s    |

PASC and ASC fields have no effect in TSB configuration.

#### NOTE

The clock frequency mentioned in the preceding table is given as an example. Refer to the clocking chapter for the frequency used to drive this module in the device.

### 39.5.5.4 Delay after Transfer ( $t_{DT}$ )

The Delay after Transfer is the minimum time between negation of the PCS signal for a frame and the assertion of the PCS signal for the next frame. See [Figure 39-51](#) for an illustration of the Delay after Transfer. The PDT and DT fields in the CTARx registers select the Delay after Transfer by the formula in the DT field description. The following table shows an example of how to compute the Delay after Transfer.

**Table 39-79. Delay after Transfer computation example**

| $f_p$   | PDT  | Prescaler | DT     | Scaler | Delay after Transfer |
|---------|------|-----------|--------|--------|----------------------|
| 100 MHz | 0b01 | 3         | 0b1110 | 32768  | 0.98 ms              |

#### NOTE

The clock frequency mentioned in the preceding table is given as an example. Refer to the clocking chapter for the frequency used to drive this module in the device.

When in Non-Continuous Clock mode the  $t_{DT}$  delay is configured according to the equation specified in the CTAR[DT] field description. When in Continuous Clock mode and TSB is not enabled, the delay is fixed at 1 SCK period.

In TSB mode the Delay after Transfer is equal to a number formed by concatenation of PDT and DT fields plus 1 of the SCK clock periods. See detailed information on [Timed Serial Bus \(TSB\)](#)".

### 39.5.5.5 Peripheral Chip Select Strobe Enable ( $\overline{PCSS}$ )

The  $\overline{PCSS}$  signal provides a delay to allow the PCS signals to settle after a transition occurs thereby avoiding glitches. When the Module is in Master mode and the PCSSE bit is set in the MCR,  $\overline{PCSS}$  provides a signal for an external demultiplexer to decode peripheral chip selects other than PCS5 into glitch-free PCS signals. The following figure shows the timing of the  $\overline{PCSS}$  signal relative to PCS signals.

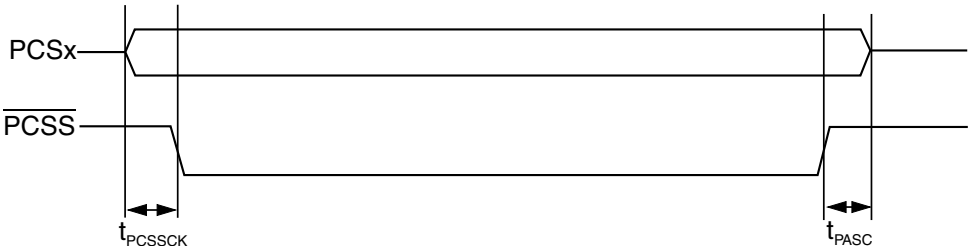


**Figure 39-50. Peripheral Chip Select Strobe timing**



The delay between the assertion of the PCS signals and the assertion of  $\overline{\text{PCSS}}$  is selected by the PCSSCK field in the CTAR based on the following formula:

$$t_{\text{PCSSCK}} = \frac{1}{f_{\text{P}}} \times \text{PCSSCK}$$

At the end of the transfer, the delay between  $\overline{\text{PCSS}}$  negation and PCS negation is selected by the PASC field in the CTAR based on the following formula:

$$t_{\text{PASC}} = \frac{1}{f_{\text{P}}} \times \text{PASC}$$

The following table shows an example of how to compute the  $t_{\text{pcssck}}$  delay.

**Table 39-80. Peripheral Chip Select Strobe Assert computation example**

| $f_{\text{P}}$ | PCSSCK | Prescaler | Delay before Transfer |
|----------------|--------|-----------|-----------------------|
| 100 MHz        | 0b11   | 7         | 70.0 ns               |

The following table shows an example of how to compute the  $t_{\text{pasc}}$  delay.

**Table 39-81. Peripheral Chip Select Strobe Negate computation example**

| $f_{\text{P}}$ | PASC | Prescaler | Delay after Transfer |
|----------------|------|-----------|----------------------|
| 100 MHz        | 0b11 | 7         | 70.0 ns              |

The  $\overline{\text{PCSS}}$  signal is not supported when Continuous Serial Communication SCK or TSB mode is enabled.

### NOTE

The clock frequency mentioned in the preceding tables is given as an example. Refer to the clocking chapter for the frequency used to drive this module in the device.

## 39.5.6 Transfer formats

The DSPI serial communication is controlled by the Serial Communications Clock (SCK) signal and the PCS signals. The SCK signal provided by the master device synchronizes shifting and sampling of the data on the SIN and SOUT pins. The PCS signals serve as enable signals for the slave devices.

In Master mode, the CPOL and CPHA bits in the Clock and Transfer Attributes Registers (CTARn) select the polarity and phase of the serial clock, SCK.

- CPOL - Selects the idle state polarity of the SCK
- CPHA - Selects if the data on SOUT is valid before or on the first SCK edge

Even though the bus slave does not control the SCK signal, in Slave mode the values of CPOL and CPHA must be identical to the master device settings to ensure proper transmission. In DSPI Slave mode, only CTAR0 is used. In DSI slave mode, only CTAR1 is used.

The module supports four different transfer formats:

- Classic DSPI with CPHA=0
- Classic DSPI with CPHA=1
- Modified Transfer Format with CPHA = 0
- Modified Transfer Format with CPHA = 1

A modified transfer format is supported to allow for high-speed communication with peripherals that require longer setup times. The module can sample the incoming data later than halfway through the cycle to give the peripheral more setup time. The MTFE bit in the MCR selects between Classic DSPI Format and Modified Transfer Format.

In the interface configurations, the module provides the option of keeping the PCS signals asserted between frames. See [Continuous Selection Format](#) for details.

### **39.5.6.1 Classic SPI Transfer Format (CPHA = 0)**

The transfer format shown in following figure is used to communicate with peripheral DSPI slave devices where the first data bit is available on the first clock edge. In this format, the master and slave sample their SIN pins on the odd-numbered SCK edges and change the data on their SOUT pins on the even-numbered SCK edges.

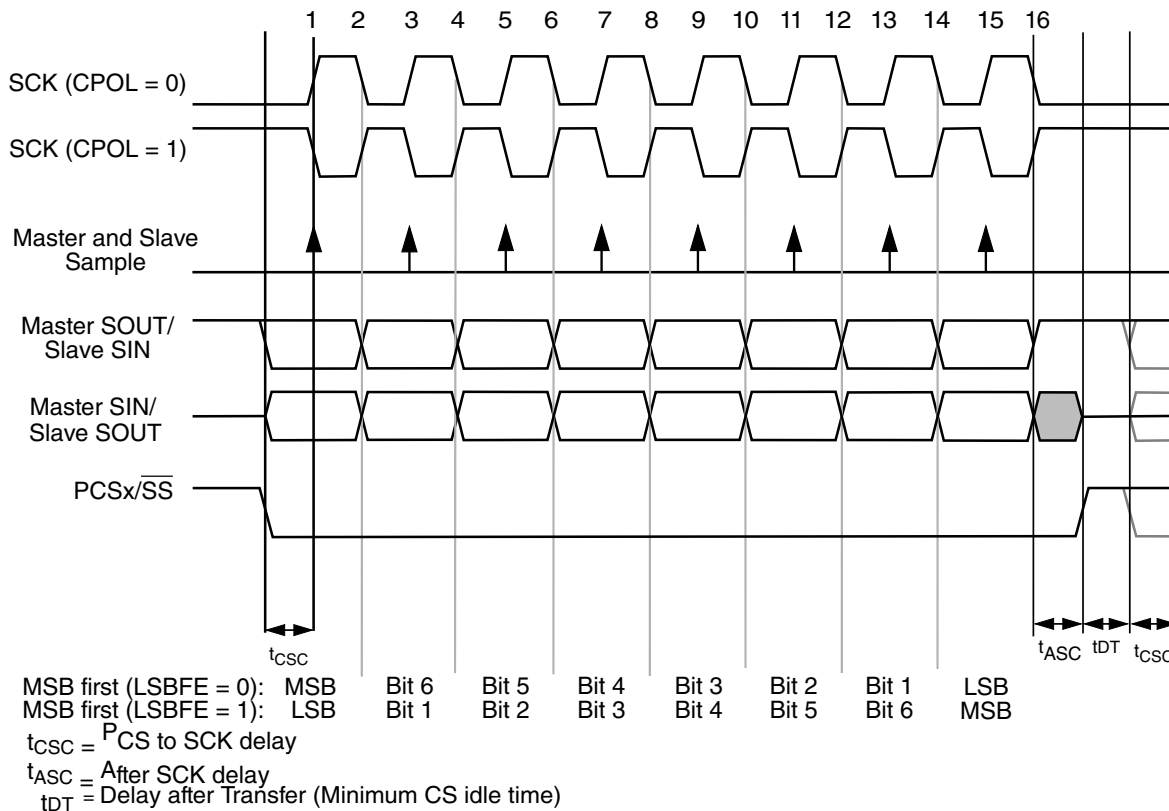


**Figure 39-51. Module transfer timing diagram (MTFE=0, CPHA=0, FMSZ=8)**

The master initiates the transfer by placing its first data bit on the SOUT pin and asserting the appropriate peripheral chip select signals to the slave device. The slave responds by placing its first data bit on its SOUT pin. After the  $t_{CSC}$  delay elapses, the master outputs the first edge of SCK. The master and slave devices use this edge to sample the first input data bit on their serial data input signals. At the second edge of the SCK, the master and slave devices place their second data bit on their serial data output signals. For the rest of the frame the master and the slave sample their SIN pins on the odd-numbered clock edges and changes the data on their SOUT pins on the even-numbered clock edges. After the last clock edge occurs, a delay of  $t_{ASC}$  is inserted before the master negates the PCS signals. A delay of  $t_{DT}$  is inserted before a new frame transfer can be initiated by the master.

### 39.5.6.2 Classic DSPI Transfer Format (CPHA = 1)

This transfer format shown in the following figure is used to communicate with peripheral DSPI slave devices that require the first SCK edge before the first data bit becomes available on the slave SOUT pin. In this format, the master and slave devices change the data on their SOUT pins on the odd-numbered SCK edges and sample the data on their SIN pins on the even-numbered SCK edges.

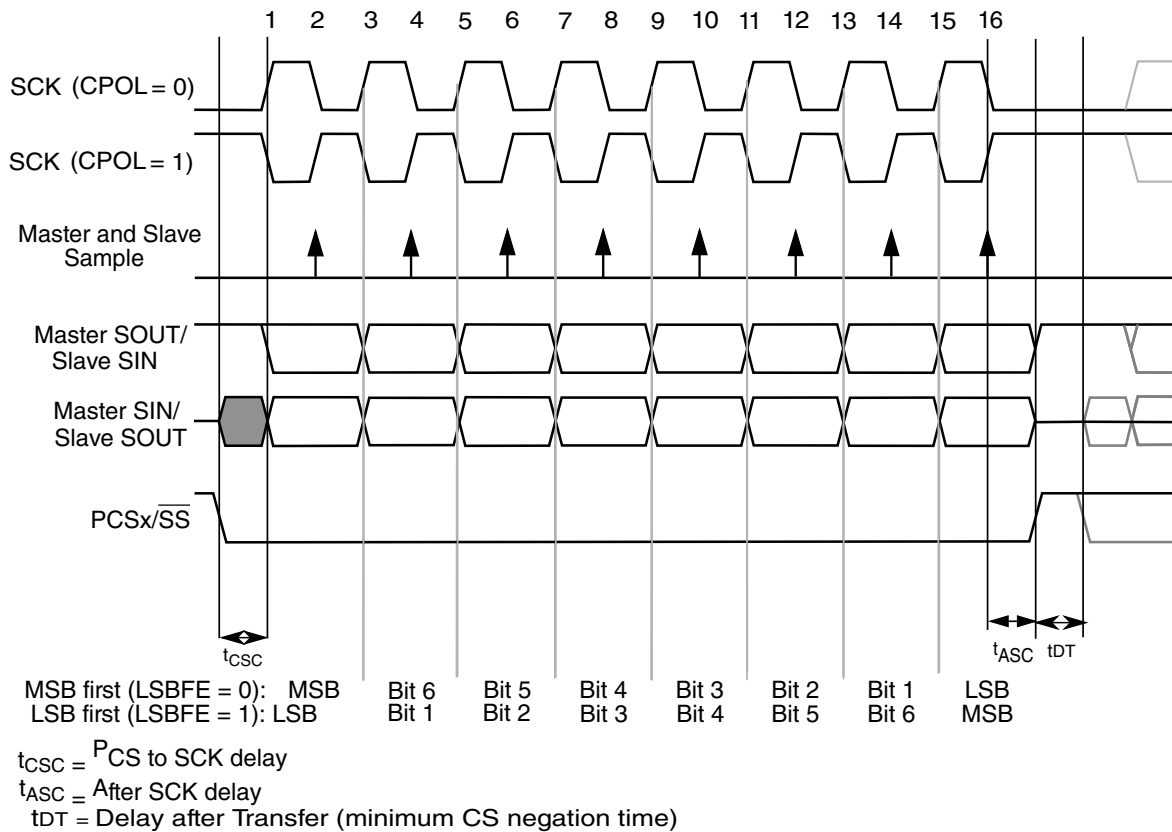


**Figure 39-52. Module transfer timing diagram (MTFE=0, CPHA=1, FMSZ=8)**

The master initiates the transfer by asserting the PCS signal to the slave. After the  $t_{CSC}$  delay has elapsed, the master generates the first SCK edge and at the same time places valid data on the master SOUT pin. The slave responds to the first SCK edge by placing its first data bit on its slave SOUT pin.

At the second edge of the SCK the master and slave sample their SIN pins. For the rest of the frame the master and the slave change the data on their SOUT pins on the odd-numbered clock edges and sample their SIN pins on the even-numbered clock edges. After the last clock edge occurs, a delay of  $t_{ASC}$  is inserted before the master negates the PCS signal. A delay of  $t_{DT}$  is inserted before a new frame transfer can be initiated by the master.

### 39.5.6.3 Modified DSPI DSI Transfer Format (MTFE = 1, CPHA = 0)

In this Modified Transfer Format both the master and the slave sample later in the SCK period than in Classic DSPI mode to allow the logic to tolerate more delays in device pads and board traces. These delays become a more significant fraction of the SCK period as the SCK period decreases with increasing baud rates.

The master and the slave place data on the SOUT pins at the assertion of the PCS signal. After the PCS to SCK delay has elapsed the first SCK edge is generated. The slave samples the master SOUT signal on every odd numbered SCK edge. The DSPI in the slave mode when the MTFE bit is set also places new data on the slave SOUT on every odd numbered clock edge. Regular external slave, configured with CPHA=0 format drives its SOUT output at every even numbered SCK clock edge.

The DSPI master places its second data bit on the SOUT line one protocol clock after odd numbered SCK edge if the protocol clock frequency to SCK frequency ratio is higher than three. If this ratio is below four the master changes SOUT at odd numbered SCK edge. The point where the master samples the SIN is selected by the DSPI\_MCR[SMPL\_PT] field. The master sample point can be delayed by one or two protocol clock cycles. The SMPL\_PT field should be set to 0 if the protocol to SCK frequency ratio is less than 4. However if this ratio is less than 4, the actual sample point is delayed by one protocol clock cycle automatically by the design.

The following timing diagrams illustrate the DSPI operation with MTFE=1. Timing delays shown are:

- $T_{csc}$  - PCS to SCK assertion delay
- $T_{acs}$  - After SCK PCS negation delay
- $T_{su_{ms}}$  - master SIN setup time
- $T_{hd_{ms}}$  - master SIN hold time
- $T_{vd_{sl}}$  - slave data output valid time, time between slave data output SCK driving edge and data becomes valid.
- $T_{su_{sl}}$  - data setup time on slave data input
- $T_{hd_{sl}}$  - data hold time on slave data input
- $T_{sys}$  - protocol clock period.

The following figure shows the modified transfer format for CPHA = 0 and Fsys/Fsck = 4. Only the condition where CPOL = 0 is illustrated. Solid triangles show the data sampling clock edges. The two possible slave behavior are shown.

- Signal, marked "SOUT of Ext Slave", presents regular DSPI slave serial output.
- Signal, marked "SOUT of DSPI Slave", presents DSPI in the slave mode with MTFE bit set.

Other MTFE = 1 diagrams show DSPI SIN input as being driven by a regular external DSPI slave, configured according DSPI master CPHA programming.

**Note**

In the following diagrams,  $f_{sys}$  represents the protocol clock frequency from which the Baud frequency  $f_{sck}$  is derived.

**NOTE**

DSPI is used for both DSPI Mode and DSI mode in the following diagrams.

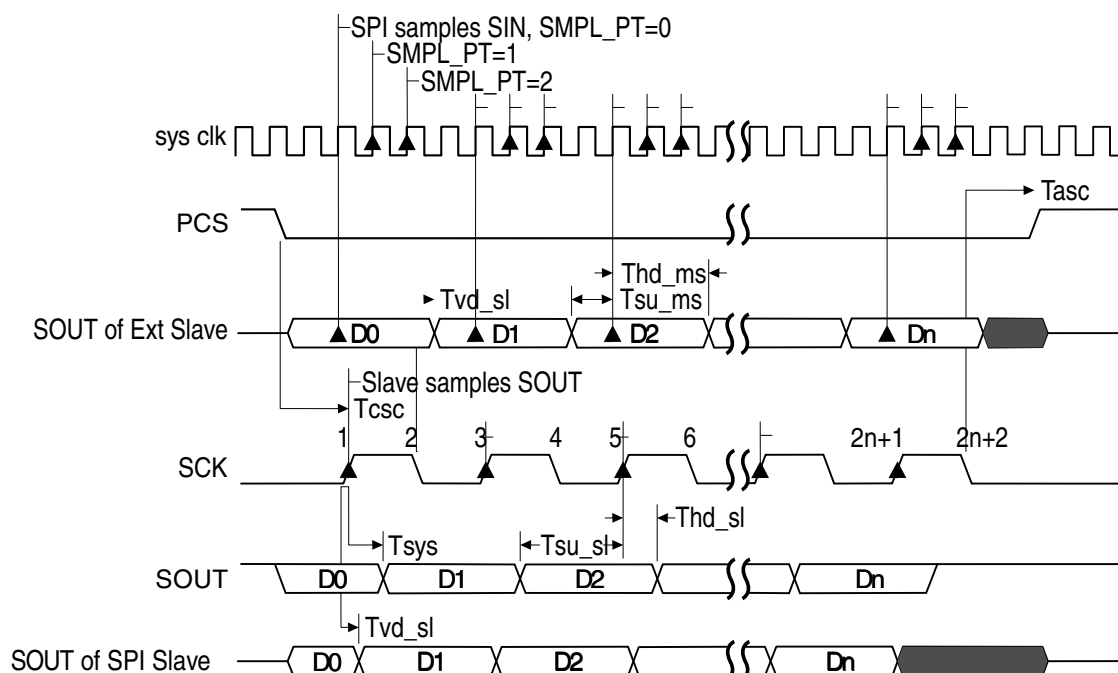


**Figure 39-53. DSPI Modified Transfer Format (MTFE=1, CPHA=0,  $f_{sck} = f_{sys}/4$ )**

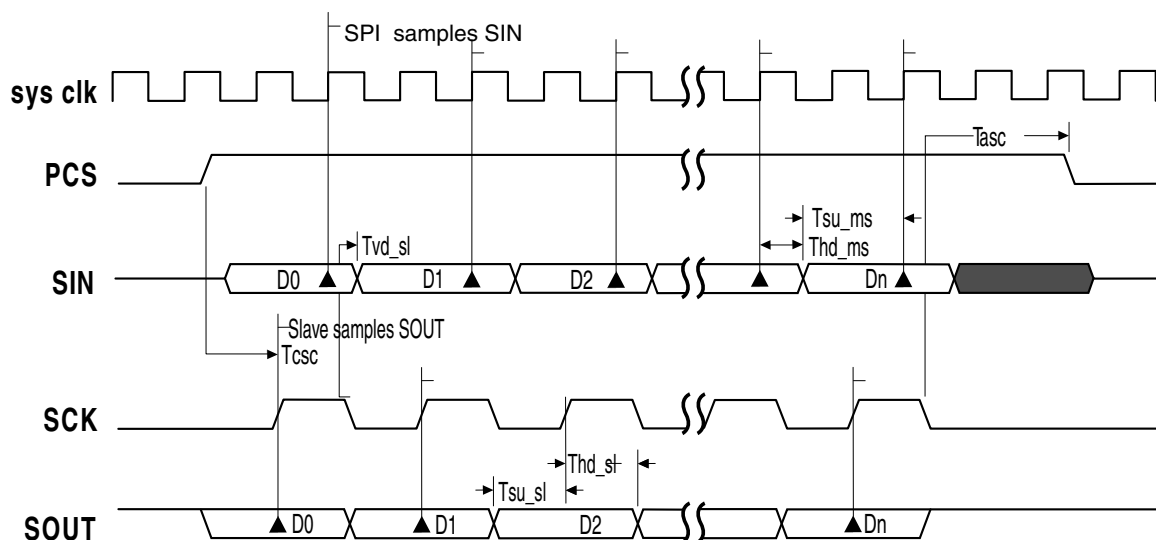


**Figure 39-54. DSPI Modified Transfer Format (MTFE=1, CPHA=0,  $f_{sck} = f_{sys}/2$ )**

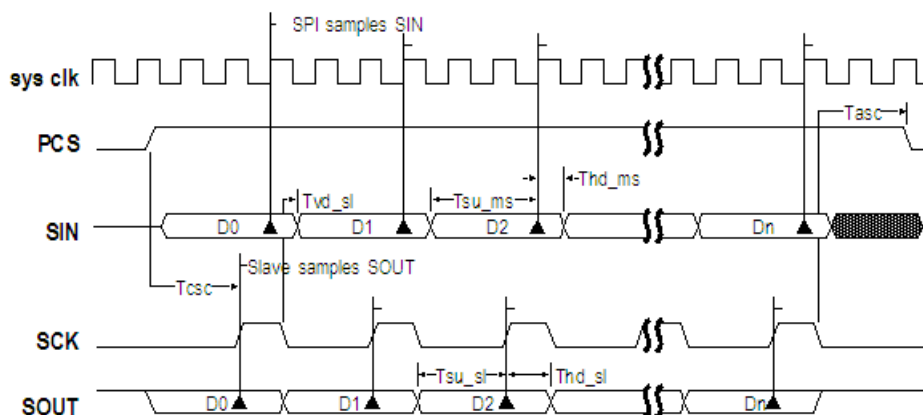


Figure 39-55. DSPI Modified Transfer Format (MTFE=1, CPHA=0,  $f_{sck} = f_{sys}/3$ )

#### 39.5.6.4 Modified SPI/DSI Transfer Format (MTFE = 1, CPHA = 1)

The following figures show the Modified Transfer Format for CPHA = 1. Only the condition, where CPOL = 0 is shown. At the start of a transfer the DSPI asserts the PCS signal to the slave device. After the PCS to SCK delay has elapsed the master and the slave put data on their SOUT pins at the first edge of SCK

#### NOTE

SPI is used for both DSPI Mode and DSI mode in the following diagrams.

. The slave samples the master SOUT signal on the even numbered edges of SCK. The master samples the slave SOUT signal on the odd numbered SCK edges starting with the third SCK edge. The slave samples the last bit on the last edge of the SCK. The master samples the last slave SOUT bit one half SCK cycle after the last edge of SCK. No clock edge will be visible on the master SCK pin during the sampling of the last bit. **The SCK to PCS delay and the After SCK delay must be greater or equal to half of the SCK period.**

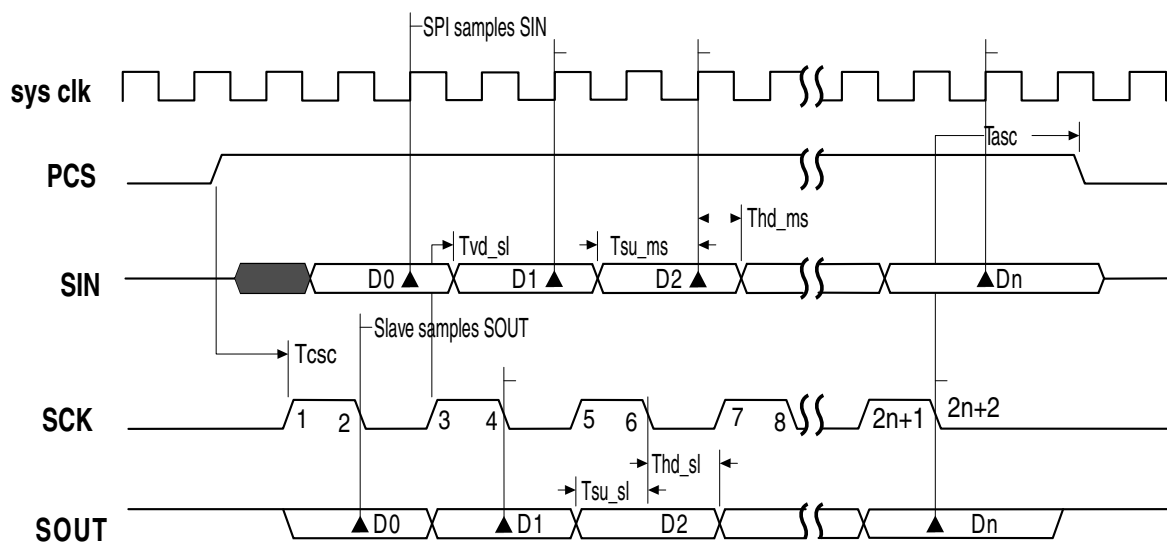


Figure 39-56. DSPI Modified Transfer Format (MTFE=1, CPHA=1,  $f_{sck} = f_{sys}/2$ )

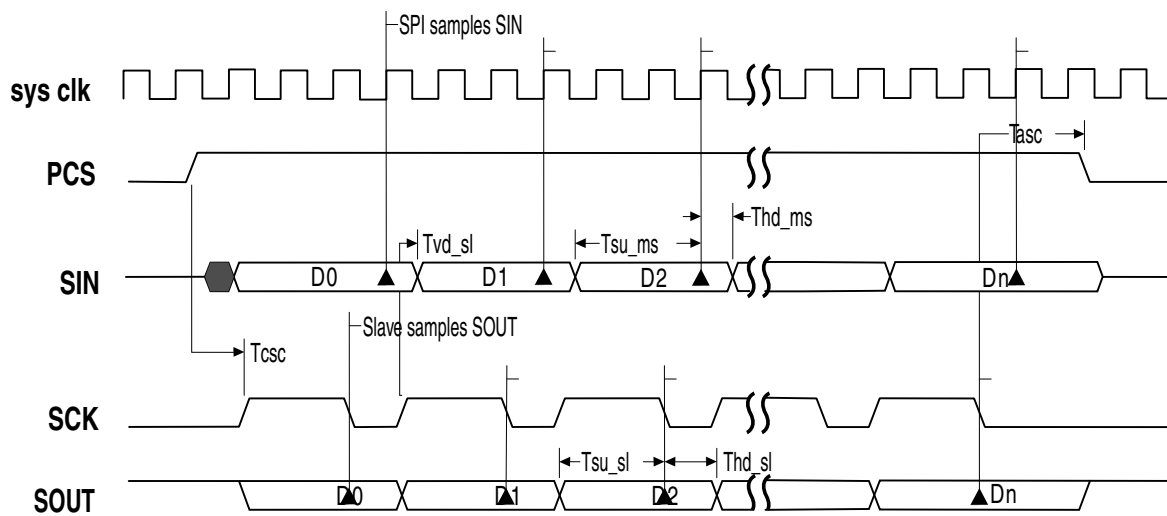


Figure 39-57. DSPI Modified Transfer Format (MTFE=1, CPHA=1,  $f_{sck} = f_{sys}/3$ )



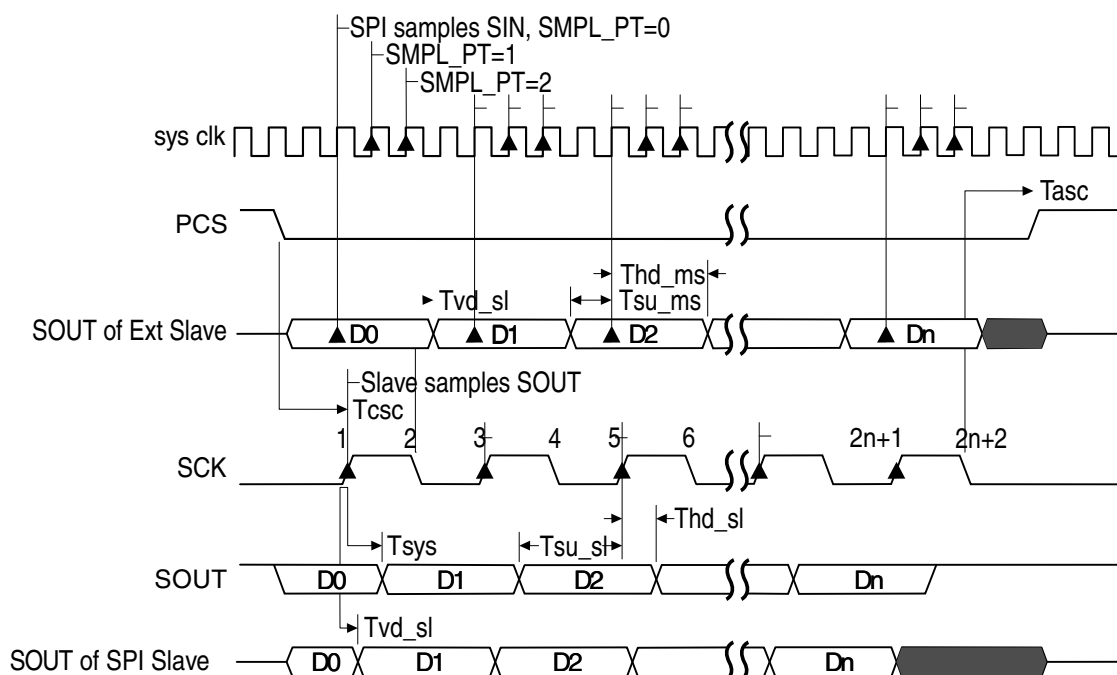


**Figure 39-58. DSPI Modified Transfer Format (MTFE=1, CPHA=1,  $f_{sck} = f_{sys}/4$ )**

### 39.5.6.5 Continuous Selection Format

Some peripherals must be deselected between every transfer. Other peripherals must remain selected between several sequential serial transfers. The Continuous Selection Format provides the flexibility to handle both the following cases. The Continuous Selection Format is enabled for the DSPI configuration by setting the CONT bit in the DSPI command. Continuous Selection is enabled for the DSI Configuration by setting the DCONT bit in the DSICR0. The behavior of the PCS signals in the two configurations is identical so only DSPI configuration will be described.

When the CONT bit = 0, the module drives the asserted Chip Select signals to their idle states in between frames. The idle states of the Chip Select signals are selected by the PCSISn bits in the MCR. The following timing diagram is for two four-bit transfers with CPHA = 1 and CONT = 0.

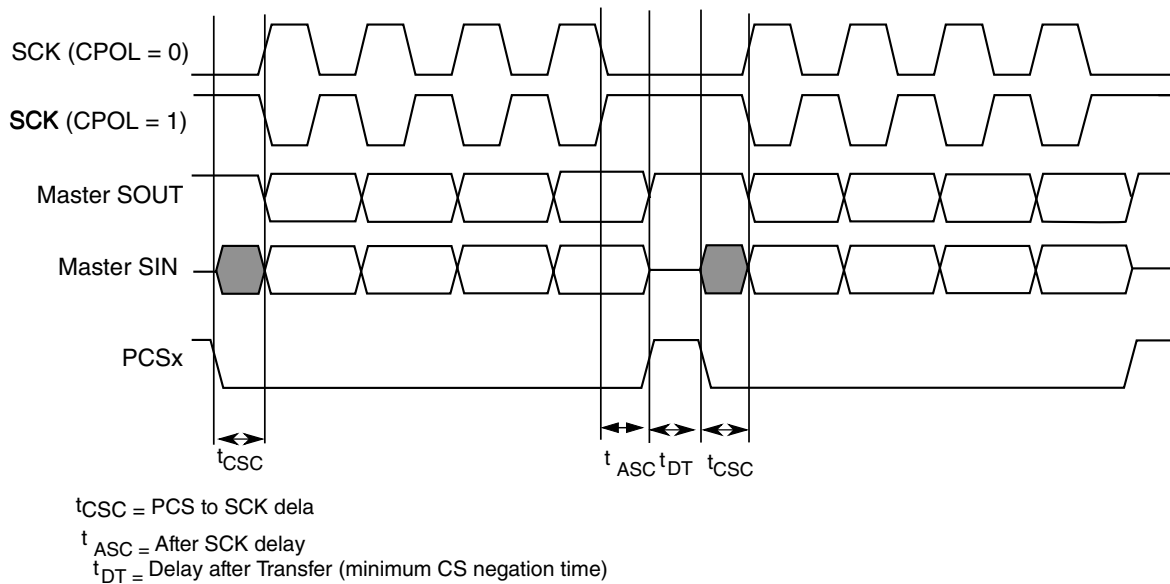


**Figure 39-59. Example of non-continuous format (CPHA=1, CONT=0)**

When the CONT bit = 1, the PCS signal remains asserted for the duration of the two transfers. The Delay between Transfers ( $t_{DT}$ ) is not inserted between the transfers. The following figure shows the timing diagram for two four-bit transfers with CPHA = 1 and CONT = 1.

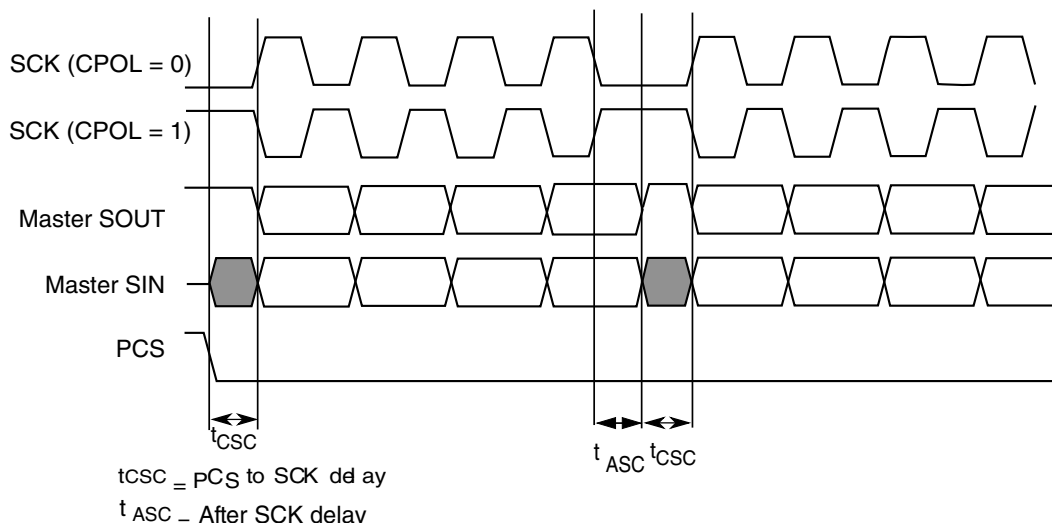


**Figure 39-60. Example of continuous transfer (CPHA=1, CONT=1)**

When using the module with continuous selection follow these rules:

- All transmit commands must have the same PCSn bits programming.
- The CTARs, selected by transmit commands, must be programmed with the same transfer attributes. Only FMSZ field can be programmed differently in these CTARs.

- When transmitting multiple frames in this mode, the user software must ensure that the last frame has the PUSHHR[CONT] bit deasserted in Master mode and the user software must provide sufficient frames in the TX\_FIFO to be sent out in Slave mode and the master deasserts the PCSn at end of transmission of the last frame.
- PUSHHR[CONT] and DSICR0[DCONT] must be deasserted before asserting MCR[HALT] in master mode. This will make sure that the PCSn signals are deasserted. Asserting MCR[HALT] during continuous transfer will cause the PCSn signals to remain asserted and hence Slave Device cannot transition from Running to Stopped state.

### NOTE

User must fill the TX FIFO with the number of entries that will be concatenated together under one PCS assertion for both master and slave before the TX FIFO becomes empty.

When operating in Slave mode, ensure that when the last entry in the TX FIFO is completely transmitted, that is, the corresponding TCF flag is asserted and TXFIFO is empty, the slave is deselected for any further serial communication; otherwise, an underflow error occurs.

## 39.5.7 Continuous Serial Communications Clock

The module provides the option of generating a Continuous SCK signal for slave peripherals that require a continuous clock.

Continuous SCK is enabled by setting the CONT\_SCKE bit in the MCR. Enabling this bit generates the Continuous SCK only if MCR[HALT] bit is low. Continuous SCK is valid in all configurations.

Continuous SCK is only supported for CPHA=1. Clearing CPHA is ignored if the CONT\_SCKE bit is set. Continuous SCK is supported for Modified Transfer Format.

Clock and transfer attributes for the Continuous SCK mode are set according to the following rules:

- When the module is in DSPI configuration, CTAR0 is used initially. At the start of each DSPI frame transfer, the CTAR specified by the CTAS for the frame is used.
- When the module is in DSI configuration, the CTAR specified by the DSICTAS field is used at all times.

- When the module is in CSI configuration, the CTAR selected by the DSICTAS field is used initially.
- At the start of a DSPI frame transfer, the CTAR specified by the CTAS value for the frame is used.
- At the start of a DSI frame transfer, the CTAR specified by the DSICTAS field is used.
- In all configurations, the currently selected CTAR remains in use until the start of a frame with a different CTAR specified, or the Continuous SCK mode is terminated.

It is recommended to keep the baud rate the same while using the Continuous SCK. Switching clock polarity between frames while using Continuous SCK can cause errors in the transfer. Continuous SCK operation is not guaranteed if the module is put into the External Stop mode or Module Disable mode.

Enabling Continuous SCK disables the PCS to SCK delay and the Delay after Transfer ( $t_{DT}$ ) is fixed to one SCK cycle. When TSB configuration is enabled the  $t_{DT}$  is programmable from 1 to 64 SCK cycles. The following figure is the timing diagram for Continuous SCK format with Continuous Selection disabled.

### NOTE

In Continuous SCK mode, for the DSPI transfer CTAR0 should always be used, and the TX FIFO must be cleared using the MCR[CLR\_TXF] field before initiating transfer.

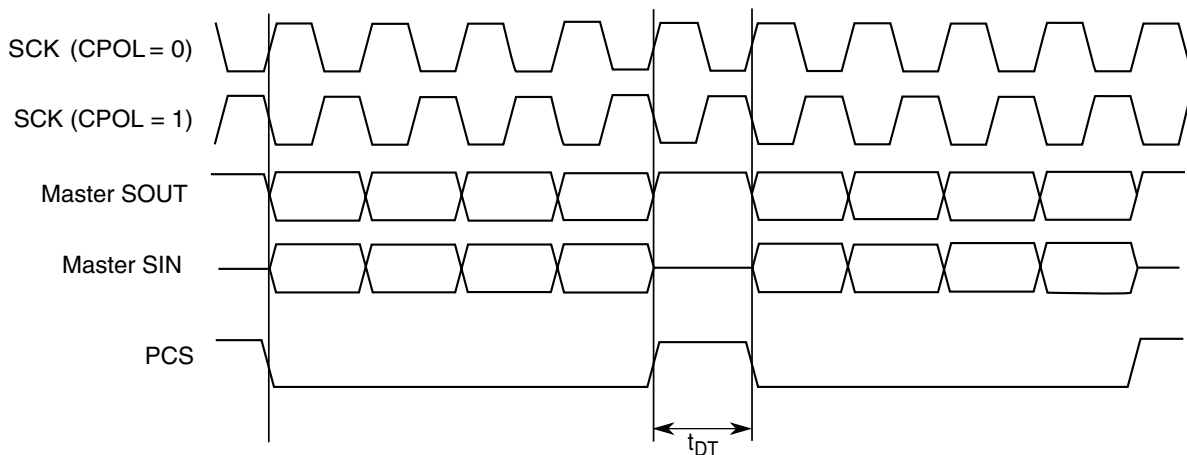


**Figure 39-61. Continuous SCK Timing Diagram (CONT=0)**

If the CONT bit in the TX FIFO entry is set or the DCONT in DSICR0 is set, PCS remains asserted between the transfers. Under certain conditions, SCK can continue with PCS asserted, but with no data being shifted out of SOUT, that is, SOUT pulled high. This can cause the slave to receive incorrect data. Those conditions include:

- Continuous SCK with CONT bit set, but no data in the TX FIFO.
- Continuous SCK with CONT bit set and entering Stopped state (refer to [Start and Stop of module transfers](#)).
- Continuous SCK with CONT bit set and entering Stop mode or Module Disable mode.

The following figure shows timing diagram for Continuous SCK format with Continuous Selection enabled.

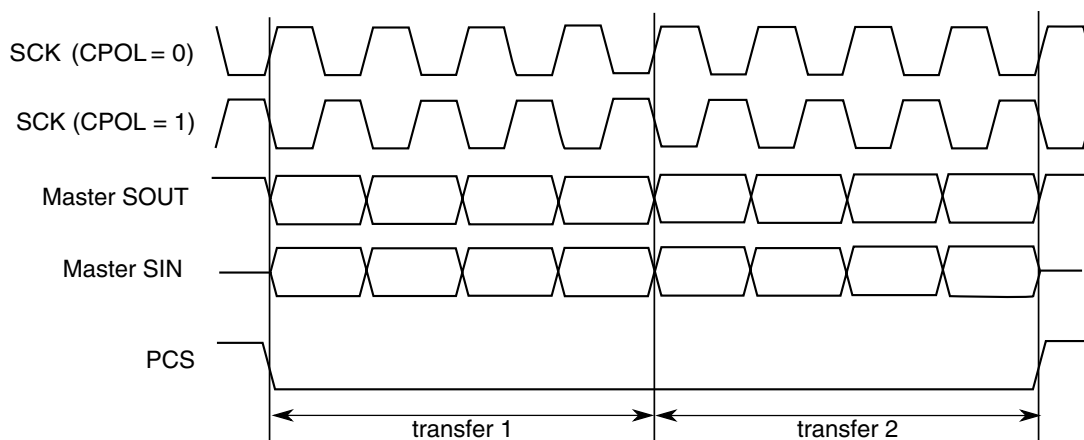


**Figure 39-62. Continuous SCK timing diagram (CONT=1)**

### 39.5.8 Slave Mode Operation Constraints

Slave mode logic shift register is buffered. This allows data streaming operation, when the module is permanently selected and data is shifted in with a constant rate.

The transmit data is transferred at second SCK clock edge of the each frame to the shift register if the  $\overline{SS}$  signal is asserted and any time when transmit data is ready and  $\overline{SS}$  signal is negated.

Received data is transferred to the receive buffer at last SCK edge of each frame, defined by frame size programmed to the CTAR0/1 register. Then the data from the buffer is transferred to the RXFIFO or DDR register.

If the  $\overline{SS}$  negates before that last SCK edge, the data from shift register is lost.

### 39.5.9 Timed Serial Bus (TSB)

The DSPI can be programmed in Timed Serial Bus configuration by setting the TSBC bit in the DSPI\_DSICR0 register. See DSPI DSI Configuration Register 0 (DSPI\_DSICR0) for details.

TSB configuration provides the Micro Second Channel (MSC) downstream channel support.

The MSC upstream channel is not supported by the DSPI, but can be supported by any available Serial Communication Controller (SCI or UART) in the SoC.

The DSI Continuous Frame Selection Format is not supported by the TSB Mode (DSICR0[DCONT] bit cannot not be asserted).

To work in TSB mode the DSPI must be in master mode and in DSI (DCONF = 0b01) or CSI (DCONF = 0b10) configuration. Both Continuous and Non Continuous Serial Communication Clock (controlled by the DSPI\_MCR[CONT\_SCKE] bit) are supported in the TSB mode.

The following figure shows the signals used in the TSB interface.

In the TSB configuration the DSPI is able to send from 4 to 34 bits MSC data frames (4 to 32 serialized data bits and up to 2 Data Selection zero bits). The serialized data bits source can be either:

- the DSPI DSI Alternate Serialization Data Register (DSPI\_ASDR0), written by the host software,
- Parallel Input pin states latched into the DSPIDSI Serialization Data Register (DSPI\_SDR0).

DSPI\_DSICR0 TXSS bit or DSPI\_SSR0 register bits define the source of the data.

The Least Significant Bits of the DSPI\_ASDR0 or DSPI\_SDR0 registers are selected to be serialized if the data frame is set to less than 32 bits.

#### NOTE

SPI is used for both DSPI Mode and DSI mode in this diagram.

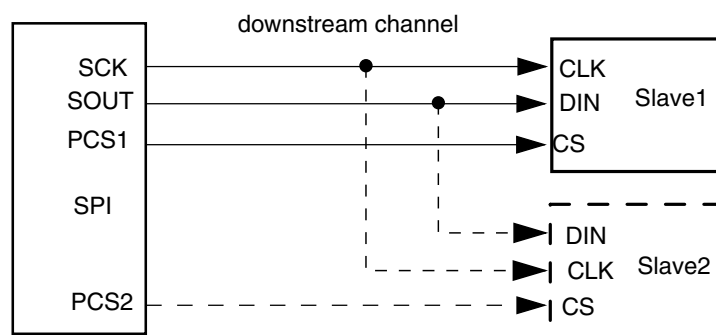


**Figure 39-63. DSPI usage in the TSB Configuration**

The PCS signals are driven together with SOUT. The  $t_{CSC}$  and  $t_{ASC}$  delays are not available. Delay after Transfer (DT) is set in SCK clock periods as a binary number formed by concatenation of the DSPI\_CTARn PDT and DT fields plus one, allowing to set DT from 1 to 64 serial clock periods. DT field provides least significant bits and PDT field provides most significant bits of the Delay after Transfer.

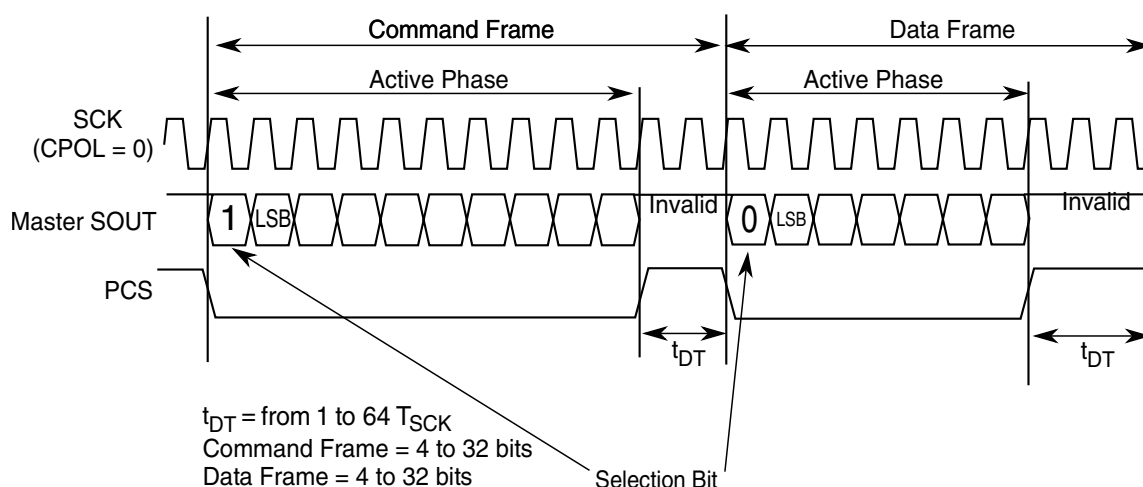


**Figure 39-64. TSB Downstream Frames**

Above figure shows the two types of MSC downstream frames - command frame, and data frame.

The first transmitted bit, called the selection bit, determines the frame type:

- The selection bit "0" indicates a data frame
- The selection bit "1" indicates a command frame

Data frame may contain up to 2 selection bits to support two external slave devices, (so called dual receiver configuration) or no selection bits at all.

The command frame can be written by software, through DSPI TX FIFO, using one or two FIFO entries with help of the CONT bit. The data frame consists of up to 32 bits from the SDR0 or ASDR0 registers and up to two selection bits (0). Number of data bits in the data frame is defined by the DSCICR1[TSBCNT] field.

The selection bit of the MSC command frames (1) can be implemented by software.

The selection bits in the data frames are enabled by DSPI\_DSICR1 DSE0 and DSE1 bits. Each DSEn bit set increases the data frame size by one bit.

To comply with MSC specification, set DSPI\_CTARn[LSBFE] to transmit least significant bit first.

Regardless the LSBFE bit setting, the Data Frame Selection Bits, if enabled, are always transmitted first, before corresponded data sub frames.

### **39.5.9.1 MSC Dual Receiver Support with PCS Switch Over**

When in TSB mode it is possible to switch the set of PCS signals that are driven during the first part of the frame to a different set of PCS signals during the second part of the frame. The bit, at which this switch over occurs, is defined by FMSZ field of the DSPI\_CTARn register, selected by DSICTAS field of the DSICR0 register.

Number of the bits, not including the Data Selection Bit, in the first part of the frame is equal to value of the FMSZ field plus one. During this part of the frame the PCS signal levels are controlled by DSPI\_DSICR0 DPCS<sub>n</sub> bits, after that by DSPI\_DSICR1 DPCS1<sub>n</sub> bits.

The PCS switch over occurs at driving edge of the SCK clock output.

The second Data Selection Bit is inserted after the PCS switch over if enabled.

Data Frame with PCS switch over is shown in the following figure.



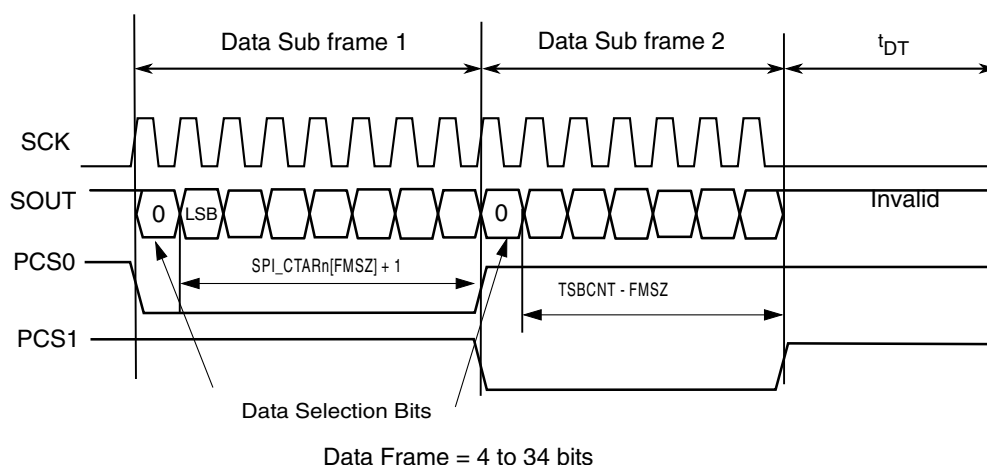


**Figure 39-65. TSB Data Frame Format for MSC Dual Receiver Operation**

### 39.5.10 Parity Generation and Check

The module can generate and check parity in the serial frame. The parity bit replaces the last transmitted bit in the frame. The parity is calculated for all transmitted data bits in frame, not including the last data bit that would be transmitted. The parity generation/control is done on frame basis. The registers field setting frame size defines the total number of bits in the frame, including the parity bit. Thus, to transmit/receive the same number of data bits with parity check, increase the frame size by one versus the same data size frame without the parity check.

Parity can be selected as odd or even. Parity Errors in the received frame set Parity Error flags in the Status register. The Parity Error Interrupt Requests are generated if enabled. The module can be programmed to stop frame transmission in case of a frame reception with parity error.

#### 39.5.10.1 Parity for DSPI Frames

When the module is in the master mode the parity generation is controlled by PE and PP bits of the CMD FIFO entries (PUSHR). Setting the PE bit enables parity generation for transmitted DSPI frames and parity check for received frames. PP bit defines polarity of the parity bit.

When continuous PCS selection is used to transmit DSPI data, two parity generation scenarios are available:

- Generate/check parity for the whole frame
- Generate/check parity for each sub-frame separately.

To generate/check parity for the whole frame set PE bit only in the last command/TX FIFO entry, forming this frame (with the PUSHR register).

To generate/check parity for each sub-frame set PE bit in each command/TX FIFO entry, forming this frame.

If the parity error occurs for received DSPI frame, the SR[SPEF] bit is set. If MCR[PES] bit is set, the module stops DSPI frames transmission. To resume DSPI operation clear the SR[SPEF] or the MCR[PES] bits.

In slave mode the parity is controlled by the PE and PP bits of the CTAR0 register similar to the master mode parity generation without continuous PCS selection.

### 39.5.10.2 Parity for DSI Frames

When DSPI is in Master Mode, parity generation is controlled by PE and PP bits of the DSPI\_DSICR0 register similar to the DSPI frames. The parity is calculated and checked for each DSI frame. DSPI\_DSICR0[DCONT] bit has no effect on parity generation. In slave mode the parity is controlled by the PE and PP bits of the CTAR1 register.

If the parity error occurs for received DSI frame, the DSPI\_SR[DPEF] bit is set. If DSPI\_DSICR0[PES] bit is set, the DSPI stops DSI frames transmission. To resume DSI operation clear the DSPI\_SR[DPEF] or the DSPI\_DSICR0[PES] bits.

### 39.5.11 Interrupts/DMA requests

The module has several conditions that can generate only interrupt requests and two conditions that can generate interrupt or DMA requests. The following table lists these conditions.

**Table 39-82. Interrupt and DMA request conditions**

| Condition                   | Flag | Interrupt | DMA |
|-----------------------------|------|-----------|-----|
| End of Queue (EOQ)          | EOQF | Yes       | -   |
| TX FIFO Fill                | TFFF | Yes       | Yes |
| Transfer Complete           | TCF  | Yes       | -   |
| TX FIFO Underflow           | TFUF | Yes       | -   |
| RX FIFO Drain               | RFDF | Yes       | Yes |
| RX FIFO Overflow            | RFOF | Yes       | -   |
| DSPI Parity Error           | SPEF | Yes       | -   |
| DSI Parity Error            | DPEF | Yes       | -   |
| DSI Deserialized Data Match | DDIF | Yes       | -   |

Each condition has a flag bit in the module Status Register (SR) and a Request Enable bit in the DMA/Interrupt Request Select and Enable Register (RSER). Certain flags (as shown in above table) generate interrupt requests or DMA requests depending on configuration of RSER register.

### 39.5.11.1 End Of Queue interrupt request

The End Of Queue (EOQ) interrupt request indicates that the end of a transmit queue is reached. The module generates the interrupt request when EOQ interrupt requests are enabled (RSER[EOQF\_RE]) and the EOQ bit in the executing DSPI command is 1.

The module generates the interrupt request when the last bit of the DSPI frame with EOQ bit set is transmitted.

### 39.5.11.2 Transmit FIFO Fill Interrupt or DMA Request

The Transmit FIFO Fill Request indicates that the TX FIFO is not full. The Transmit FIFO Fill Request is generated when the number of entries in the TX FIFO is less than the maximum number of possible entries, and the TFFF\_RE bit in the RSER is set. The TFFF\_DIRS bit in the RSER selects whether a DMA request or an interrupt request is generated.

#### NOTE

TFFF flag clears automatically when DMA is used to fill TX FIFO. Configure the DMA to fill only one FIFO location per transfer.

To clear TFFF when not using DMA, follow these steps for every PUSH performed using CPU to fill TX FIFO:

1. Wait until TFFF = 1.
2. Write data to PUSHR using CPU.
3. Clear TFFF by writing a 1 to its location. If TX FIFO is not full, this flag will not clear.

### 39.5.11.3 Transfer Complete Interrupt Request

The Transfer Complete Request indicates the end of the transfer of a serial frame. The Transfer Complete Request is generated at the end of each frame transfer when the TCF\_RE bit is set in the RSER.

#### **39.5.11.4 Transmit FIFO Underflow Interrupt Request**

The Transmit FIFO Underflow Request indicates that an underflow condition in the TX FIFO has occurred. The transmit underflow condition is detected only for the module operating in Slave mode and DSPI configuration. The TFUF bit is set when the TX FIFO of the module is empty, and a transfer is initiated from an external SPI master. If the TFUF bit is set while the TFUF\_RE bit in the RSER is set, an interrupt request is generated.

#### **39.5.11.5 Receive FIFO Drain Interrupt or DMA Request**

The Receive FIFO Drain Request indicates that the RX FIFO is not empty. The Receive FIFO Drain Request is generated when the number of entries in the RX FIFO is not zero, and the RFDF\_RE bit in the RSER is set. The RFDF\_DIRS bit in the RSER selects whether a DMA request or an interrupt request is generated. Configure the DMA to drain only one FIFO location per transfer.

#### **39.5.11.6 Receive FIFO Overflow Interrupt Request**

The Receive FIFO Overflow Request indicates that an overflow condition in the RX FIFO has occurred. A Receive FIFO Overflow request is generated when RX FIFO and shift register are full and a transfer is initiated. The RFOF\_RE bit in the RSER must be set for the interrupt request to be generated.

Depending on the state of the ROOE bit in the MCR, the data from the transfer that generated the overflow is either ignored or shifted in to the shift register. If the ROOE bit is set, the incoming data is shifted in to the shift register. If the ROOE bit is cleared, the incoming data is ignored.

#### **39.5.11.7 DSPI Frame Parity Error Interrupt Request**

The SPI Frame Parity Error Flag indicates that a DSPI frame with parity error had been received. The SPEF\_RE bit in the RSER must be set for the interrupt request to be generated.

### 39.5.11.8 DSI Frame Parity Error Interrupt Request

The DSI Frame Parity Error Flag indicates that a DSI frame with parity error has been received. The DPEF\_RE bit in the DSPI\_RSER must be set for the interrupt request to be generated.

### 39.5.11.9 Deserialized Data Match Interrupt Request

The Deserialized Data Match Flag (DDIF) indicates that a DSI frame matches DSPI\_DPIR0 data, masked with DSPI\_DIMR0, had been received. The DDIF\_RE bit in the DSPI\_RSER must be set for the request to be generated.

## 39.5.12 Power saving features

The module supports following power-saving strategies:

- External Stop mode
- Module Disable mode – Clock gating of non-memory mapped logic

### 39.5.12.1 Stop mode (External Stop mode)

This module supports the Stop mode protocol. When a request is made to enter External Stop mode, the module acknowledges the request. If a serial transfer is in progress, then this module waits until it reaches the frame boundary before it is ready to have its clocks shut off. While the clocks are shut off, this module's memory-mapped logic is not accessible. This also puts the module in STOPPED state. The SR[TXRXS] bit is cleared to indicate STOPPED state. The states of the interrupt and DMA request signals cannot be changed while in External Stop mode.

### 39.5.12.2 Module Disable mode

Module Disable mode is a block-specific mode that the module can enter to save power. Host CPU can initiate the Module Disable mode by setting the MDIS bit in the MCR. The Module Disable mode can also be initiated by hardware.

When the MDIS bit is set, the module negates the Clock Enable signal at the next frame boundary. Once the Clock Enable signal is negated, it is said to have entered Module Disable Mode. This also puts the module in STOPPED state. The SR[TXRXS] bit is

cleared to indicate STOPPED state. If implemented, the Clock Enable signal can stop the clock to the non-memory mapped logic. When Clock Enable is negated, the module is in a dormant state, but the memory mapped registers are still accessible. Certain read or write operations have a different effect when the module is in the Module Disable mode. Reading the RX FIFO Pop Register does not change the state of the RX FIFO. Similarly, writing to the PUSHHR Register does not change the state of the TX FIFO or CMD FIFO. Clearing either of the FIFOs has no effect in the Module Disable mode. Changes to the DIS\_TXF and DIS\_RXF fields of the MCR have no effect in the Module Disable mode. In the Module Disable mode, all status bits and register flags in the module return the correct values when read, but writing to them has no effect. Writing to the TCR during Module Disable mode has no effect. Interrupt and DMA request signals cannot be cleared while in the Module Disable mode.

## 39.6 Initialization/application information

This section describes how to initialize the module.

### 39.6.1 How to manage queues

The queues are not part of the module, but it includes features in support of queue management. Queues are primarily supported in DSPI configuration.

1. When module executes last command word from a queue, the EOQ bit in the command word is set to indicate it that this is the last entry in the queue.
2. At the end of the transfer, corresponding to the command word with EOQ set is sampled, the EOQ flag (EOQF) in the SR is set.
3. The setting of the EOQF flag disables serial transmission and reception of data, putting the module in the Stopped state. The TXRXS bit is cleared to indicate the Stopped state.
4. The DMA can continue to fill TX FIFO until it is full or step 5 occurs.
5. Disable DMA transfers by disabling the DMA enable request for the DMA channel assigned to TX FIFO (and CMD FIFO) and RX FIFO. This is done by clearing the corresponding DMA enable request bits in the DMA Controller.
6. Ensure all received data in RX FIFO has been transferred to memory receive queue by reading the RXCNT in SR or by checking RFDF in the SR after each read operation of the POPR.

7. Modify DMA descriptor of TX and RX channels for new queues
8. Flush TX FIFO (and CMD FIFO) by writing a 1 to the CLR\_TXF bit in the MCR. Flush RX FIFO by writing a '1' to the CLR\_RXF bit in the MCR.
9. Clear transfer count either by setting CTCNT bit in the command word of the first entry in the new queue or via CPU writing directly to DSPI\_TCNT field in the TCR.
10. Enable DMA channel by enabling the DMA enable request for the DMA channel assigned to the module TX FIFO, (and CMD FIFO) and RX FIFO by setting the corresponding DMA set enable request bit.
11. Enable serial transmission and serial reception of data by clearing the EOQF bit.

### 39.6.2 Switching Master and Slave mode

When changing modes in the module, follow the steps below to guarantee proper operation.

1. Halt it by setting MCR[HALT].
2. Clear the transmit and receive FIFOs by writing a 1 to the CLR\_TXF and CLR\_RXF bits in MCR.
3. Set the appropriate mode in MCR[MSTR] and enable it by clearing MCR[HALT].

### 39.6.3 Initializing Module in Master/Slave Modes

Once the appropriate mode in MCR[MSTR] is configured, the module is enabled by clearing MCR[HALT]. It should be ensured that module Slave is enabled before enabling it's Master. This ensures the Slave is ready to be communicated with, before Master initializes communication.

### 39.6.4 Baud rate settings

The following table shows the baud rate that is generated based on the combination of the baud rate prescaler PBR and the baud rate scaler BR in the CTARs. The values calculated assume a 100 MHz protocol frequency and the double baud rate DBR bit is cleared.

**Table 39-83. Baud rate values (bps)**

|                         |       | Baud rate divider prescaler values |       |       |       |
|-------------------------|-------|------------------------------------|-------|-------|-------|
|                         |       | 2                                  | 3     | 5     | 7     |
| Baud Rate Scaler Values | 2     | 25.0M                              | 16.7M | 10.0M | 7.14M |
|                         | 4     | 12.5M                              | 8.33M | 5.00M | 3.57M |
|                         | 6     | 8.33M                              | 5.56M | 3.33M | 2.38M |
|                         | 8     | 6.25M                              | 4.17M | 2.50M | 1.79M |
|                         | 16    | 3.12M                              | 2.08M | 1.25M | 893k  |
|                         | 32    | 1.56M                              | 1.04M | 625k  | 446k  |
|                         | 64    | 781k                               | 521k  | 312k  | 223k  |
|                         | 128   | 391k                               | 260k  | 156k  | 112k  |
|                         | 256   | 195k                               | 130k  | 78.1k | 55.8k |
|                         | 512   | 97.7k                              | 65.1k | 39.1k | 27.9k |
|                         | 1024  | 48.8k                              | 32.6k | 19.5k | 14.0k |
|                         | 2048  | 24.4k                              | 16.3k | 9.77k | 6.98k |
|                         | 4096  | 12.2k                              | 8.14k | 4.88k | 3.49k |
|                         | 8192  | 6.10k                              | 4.07k | 2.44k | 1.74k |
|                         | 16384 | 3.05k                              | 2.04k | 1.22k | 872   |
|                         | 32768 | 1.53k                              | 1.02k | 610   | 436   |

### 39.6.5 Delay settings

The following table shows the values for the Delay after Transfer ( $t_{DT}$ ) and CS to SCK Delay ( $T_{CSC}$ ) that can be generated based on the prescaler values and the scaler values set in the CTARs. The values calculated assume a 100 MHz protocol frequency.

#### NOTE

The clock frequency mentioned above is given as an example in this chapter. See the clocking chapter for the frequency used to drive this module in the device.

This table does not apply for TSB Continuous mode.



**Table 39-84. Delay values**

|                     |       | Delay prescaler values |               |               |               |
|---------------------|-------|------------------------|---------------|---------------|---------------|
|                     |       | 1                      | 3             | 5             | 7             |
| Delay scaler values | 2     | 20.0 ns                | 60.0 ns       | 100.0 ns      | 140.0 ns      |
|                     | 4     | 40.0 ns                | 120.0 ns      | 200.0 ns      | 280.0 ns      |
|                     | 8     | 80.0 ns                | 240.0 ns      | 400.0 ns      | 560.0 ns      |
|                     | 16    | 160.0 ns               | 480.0 ns      | 800.0 ns      | 1.1 $\mu$ s   |
|                     | 32    | 320.0 ns               | 960.0 ns      | 1.6 $\mu$ s   | 2.2 $\mu$ s   |
|                     | 64    | 640.0 ns               | 1.9 $\mu$ s   | 3.2 $\mu$ s   | 4.5 $\mu$ s   |
|                     | 128   | 1.3 $\mu$ s            | 3.8 $\mu$ s   | 6.4 $\mu$ s   | 9.0 $\mu$ s   |
|                     | 256   | 2.6 $\mu$ s            | 7.7 $\mu$ s   | 12.8 $\mu$ s  | 17.9 $\mu$ s  |
|                     | 512   | 5.1 $\mu$ s            | 15.4 $\mu$ s  | 25.6 $\mu$ s  | 35.8 $\mu$ s  |
|                     | 1024  | 10.2 $\mu$ s           | 30.7 $\mu$ s  | 51.2 $\mu$ s  | 71.7 $\mu$ s  |
|                     | 2048  | 20.5 $\mu$ s           | 61.4 $\mu$ s  | 102.4 $\mu$ s | 143.4 $\mu$ s |
|                     | 4096  | 41.0 $\mu$ s           | 122.9 $\mu$ s | 204.8 $\mu$ s | 286.7 $\mu$ s |
|                     | 8192  | 81.9 $\mu$ s           | 245.8 $\mu$ s | 409.6 $\mu$ s | 573.4 $\mu$ s |
|                     | 16384 | 163.8 $\mu$ s          | 491.5 $\mu$ s | 819.2 $\mu$ s | 1.1 ms        |
|                     | 32768 | 327.7 $\mu$ s          | 983.0 $\mu$ s | 1.6 ms        | 2.3 ms        |
|                     | 65536 | 655.4 $\mu$ s          | 2.0 ms        | 3.3 ms        | 4.6 ms        |

### 39.6.6 Calculation of FIFO pointer addresses

Complete visibility of the FIFO contents is available through the FIFO registers, and valid entries can be identified through a memory-mapped pointer and counter for each FIFO. The pointer to the first-in entry in each FIFO is memory mapped. For the TX FIFO the first-in pointer is the Transmit Next Pointer (TXNXTPTR). For the CMD FIFO the first-in pointer is the Command Next Pointer (CMDNXTPTR). For the RX FIFO the first-in pointer is the Pop Next Pointer (POPNXTPTR). The following figure illustrates the concept of first-in and last-in FIFO entries along with the FIFO Counter. The TX FIFO is chosen for the illustration, but the concepts carry over. See [Transmit First In First Out \(TX FIFO\) buffering mechanism](#), [Command First In First Out \(CMD FIFO\) Buffering Mechanism](#) and [Receive First In First Out \(RX FIFO\) buffering mechanism](#) for details on the FIFO operation.

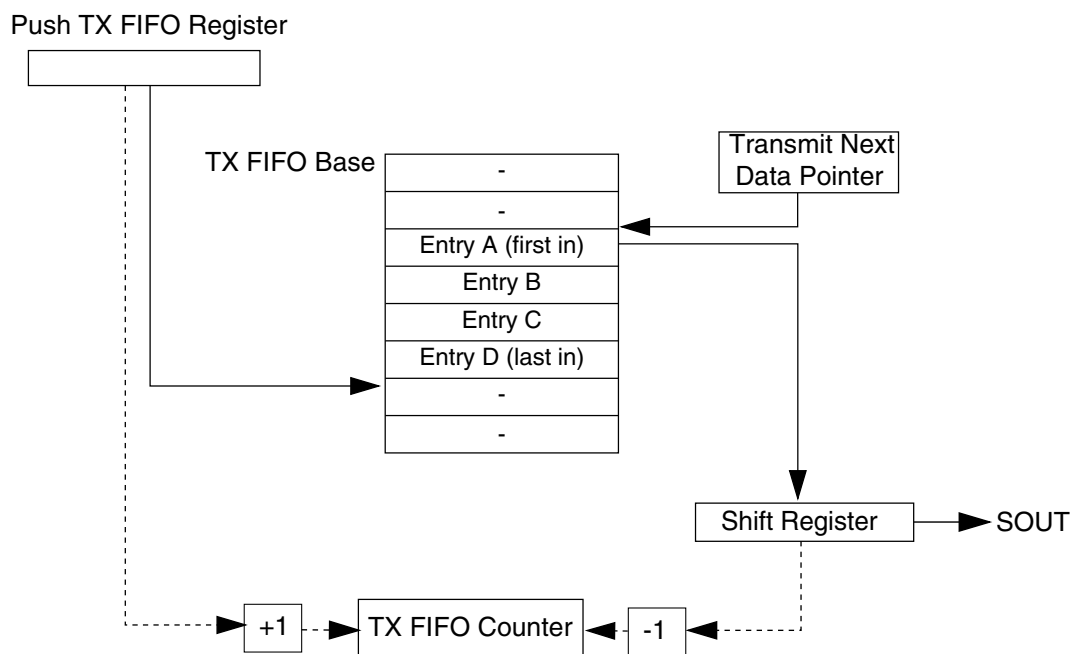


Figure 39-66. TX FIFO pointers and counter

### 39.6.6.1 Address Calculation for the First-in Entry and Last-in Entry in the TX FIFO

The memory address of the first-in entry in the TX FIFO is computed by the following equation:

$$\text{First-in EntryAddress} = \text{TXFIFOBBase} + (4 \times \text{TXNXTPTR})$$

The memory address of the last-in entry in the TX FIFO is computed by the following equation:

$$\text{Last-inEntryaddress} = \text{TXFIFOBBase} + 4 \times (\text{TXCTR} + \text{TXNXTPTR} - 1) \bmod (\text{TXFIFOdepth})$$

TX FIFO Base - Base address of TX FIFO

TXCTR - TX FIFO Counter

TXNXTPTR - Transmit Next Pointer

TX FIFO Depth - Transmit FIFO depth, implementation specific

### 39.6.6.2 Address Calculation for the First-in Entry and Last-in Entry in the CMD FIFO

The memory address of the first-in entry in the CMD FIFO is computed by the following equation:

$$\text{First-in EntryAddress} = \text{TXFIFOBase} + (4 \times \text{TXNXTPTR})$$

The memory address of the last-in entry in the CMD FIFO is computed by the following equation:

$$\text{Last-inEntryaddress} = \text{TXFIFOBase} + 4 \times (\text{TXCTR} + \text{TXNXTPTR} - 1) \bmod (\text{TXFIFOdepth})$$

CMD FIFO Base - Base address of CMD FIFO

CMDCTR - CMD FIFO Counter

CMDNXTPTR - Command Next Pointer

CMD FIFO Depth - Command FIFO depth, implementation specific

### 39.6.6.3 Address Calculation for the First-in Entry and Last-in Entry in the RX FIFO

The memory address of the first-in entry in the RX FIFO is computed by the following equation:

$$\text{First-in EntryAddress} = \text{RX FIFOBase} + (4 \times \text{POPNXTPTR})$$

The memory address of the last-in entry in the RX FIFO is computed by the following equation:

$$\text{Last-inEntryaddress} = \text{RX FIFO Base} + 4 \times (\text{RXCTR} + \text{POPNXTPTR} - 1) \bmod (\text{RXFIFOdepth})$$

RX FIFO Base - Base address of RX FIFO

RXCTR - RX FIFO counter

POPNXTPTR - Pop Next Pointer

RX FIFO Depth - Receive FIFO depth, implementation specific



## Chapter 40

# Fast Ethernet Controller (FEC)

### 40.1 Chip-specific FEC information

This chip has one instance of the FEC module.

#### 40.1.1 FEC interface selection

On this chip, the FEC can operate using MII-Lite or RMII interfaces.

#### 40.1.2 FEC clocking

See [Ethernet clocking](#) for information about FEC clocking restrictions on this chip.

### 40.2 Introduction

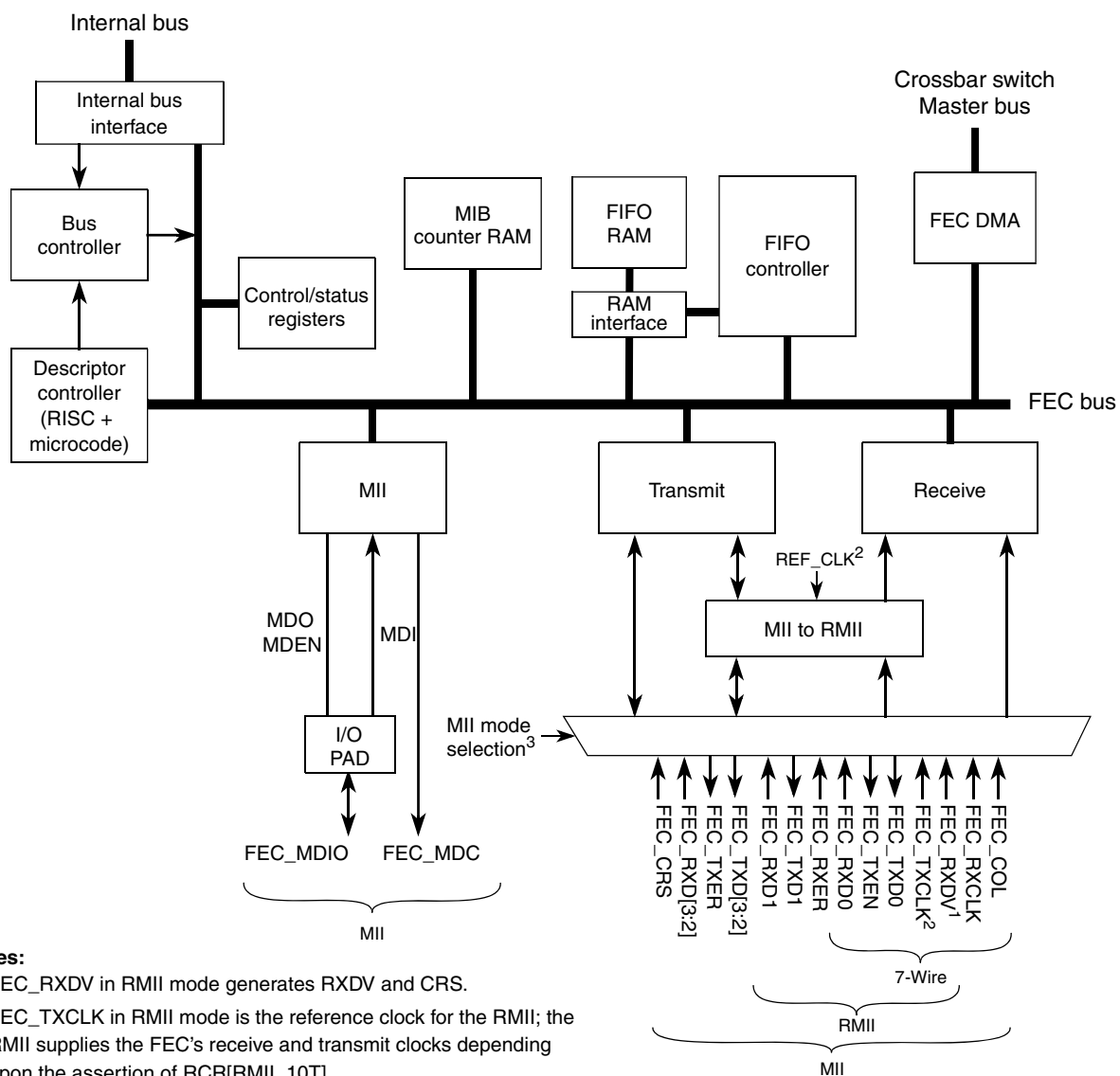
The Fast Ethernet Controller (FEC) is a communication controller that supports 10 and 100 Mbps Ethernet/IEEE 802.3 networks. An external transceiver interface and transceiver function are required to complete the interface to the media. The FEC supports five different standard MAC-PHY (physical) interfaces for connection to an external Ethernet transceiver. The FEC supports the 10/100 Mbps MII, 10/100 Mbps MII-Lite, 10/100 Mbps reduced MII, and the 10 Mbps-only 7-wire interface.

#### 40.2.1 Block diagram

[Figure 40-1](#) shows the block diagram of the FEC. The FEC is implemented with a combination of hardware and microcode. The off-chip (Ethernet) interfaces are compliant with industry and IEEE 802.3 standards.

## NOTE

When MII-Lite is used on this chip, the FEC\_TXER, FEC\_RXER, FEC\_COL, and FEC\_CRS signals shown in [Figure 40-1](#) are not used.



### Figure 40-1. FEC block diagram

The following table provides a brief description of the blocks in the previous figure.

### Table 40-1. FEC sub-block descriptions

| Block                              | Description                                                                          |
|------------------------------------|--------------------------------------------------------------------------------------|
| Control and status registers (CSR) | Provides global control (Ethernet reset and enable) and interrupt managing registers |

*Table continues on the next page...*

**Table 40-1. FEC sub-block descriptions (continued)**

| Block                             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Descriptor controller             | <p>A RISC-based controller providing these functions in the FEC:</p> <ul style="list-style-type: none"> <li>• Initialization (those internal registers not initialized by you or hardware)</li> <li>• High-level control of the DMA channels (initiating DMA transfers)</li> <li>• Interpreting buffer descriptors</li> <li>• Address recognition for receive frames</li> <li>• Random number generation for transmit collision backoff timer</li> </ul> |
| DMA block                         | <p>Provides multiple channels allowing transmit data, transmit descriptor, receive data and receive descriptor accesses to all run independently</p> <p><b>Note:</b> In this document, "DMA" refers to the DMA block within the FEC, and is not related to any other DMA controllers that may be present on the chip.</p>                                                                                                                                |
| Media independent interface (MII) | <p>Provides a serial channel for control/status communication with the external physical layer device (transceiver). This serial channel consists of:</p> <ul style="list-style-type: none"> <li>• The management data clock (FEC_MDC)</li> <li>• The management data input/output lines (FEC_MDIO) of the MII interface</li> </ul>                                                                                                                      |
| Message information block (MIB)   | <p>Maintains counters for a variety of network events and statistics. It is not necessary for operation of the FEC, but provides valuable counters for network management.</p> <p>The counters supported are:</p> <ul style="list-style-type: none"> <li>• The RMON (RFC 1757) Ethernet Statistics group</li> <li>• Some of the IEEE 802.3 counters</li> </ul>                                                                                           |
| RAM                               | <p>Serves as the focal point of all data flow in the Fast Ethernet controller and divides into transmit and receive FIFOs. The FIFO boundaries are programmable using the FRSR register.</p> <p>User data flows to/from the DMA block from/to the receive/transmit FIFOs.</p> <p>Transmit data flows from the transmit FIFO into the transmit block.</p> <p>Receive data flows from the receive block into the receive FIFO.</p>                         |
| Transmit and receive blocks       | Provide the Ethernet MAC functionality (with some assist from microcode)                                                                                                                                                                                                                                                                                                                                                                                 |

## 40.2.2 Features

The FEC incorporates the following features:

- Support for the following Ethernet physical interfaces:<sup>1</sup>
  - 100-Mbps IEEE 802.3 MII
  - 10-Mbps IEEE 802.3 MII
  - 100-Mbps IEEE 802.3 MII-Lite

1. For interface selection options, see the chip-specific FEC information.

- 10-Mbps IEEE 802.3 MII-Lite
- 100-Mbps reduced media independent interface (RMII)
- 10-Mbps reduced media independent interface (RMII)
- 10-Mbps 7-wire interface (industry standard)
- IEEE 802.3 full duplex flow control
- Programmable max frame length supports IEEE 802.1 VLAN tags and priority
- Support for full-duplex operation (200 Mbps throughput) with a minimum internal bus clock rate of 50 MHz
- Support for half-duplex operation (100 Mbps throughput) with a minimum internal bus clock rate of 25 MHz
- Retransmission from transmit FIFO following a collision (no processor bus utilization)
- Automatic internal flushing of the receive FIFO for runts (collision fragments) and address recognition rejects (no processor bus utilization)
- Address recognition
  - Frames with broadcast address may be always accepted or always rejected
  - Exact match for single 48-bit individual (unicast) address
  - Hash (64-bit hash) check of individual (unicast) addresses
  - Hash (64-bit hash) check of group (multicast) addresses
  - Promiscuous mode

### **40.2.3 Clocking requirements**

The FEC system clock frequency must be equal to or greater than the FEC protocol reference clock frequency (for example, 25 MHz system frequency for 100 Mbps protocol frequency). This requirement is true regardless of the reference selected by the application.

## **40.3 Modes of operation**

This section describes the FEC operating modes.



### 40.3.1 Full and half duplex operation

Full duplex mode is for use on point-to-point links between switches or end node to switch. Half duplex mode works in connections between an end node and a repeater or between repeaters. TCR[FDEN] controls duplex mode selection.

When configured for full duplex mode, flow control may be enabled. For more details, see:

- The RFC\_PAUSE and TFC\_PAUSE fields in the TCR
- The RCR[FCE] field
- [Full duplex flow control](#)

### 40.3.2 Interface options

The following interface options are supported<sup>2</sup>. A detailed discussion of the interface configurations is provided in [Network interface options](#).

#### 40.3.2.1 10 Mbps and 100 Mbps MII interface

The IEEE 802.3 standard defines the media independent interface (MII) for 10/100 Mbps operation. The MAC-PHY interface may be configured to operate in MII mode by programming RCR[MII\_MODE]=1.

The operation speed is determined by the FEC\_TXCLK and FEC\_RXCLK pins driven by the external transceiver. The transceiver auto-negotiates the speed or software controls it via the serial management interface (FEC\_MDC/FEC\_MDIO pins) to the transceiver. Refer to the MMFR and MSCR register descriptions, as well as the section on the MII, for a description of how to read and write registers in the transceiver via this interface.

#### 40.3.2.2 10 Mbps and 100 Mbps MII-Lite interface

MII-Lite refers to MII without the following signals:

- FEC\_COL
- FEC\_CRs

---

2. For interface selection options, see the chip-specific FEC information.

- FEC\_RXER
- FEC\_TXER

The MII-Lite connection is for PHYs that do not have all the MII pins and signals that are available for MII.

The following table provides additional information on the behavior of these signals when MII-Lite is used.

**Table 40-2. MII signals not used in MII-Lite**

| Signal   | Information                                                                                                                                                                                                                                                                                                                                                                                                                           |
|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| FEC_COL  | This signal is asserted by the PHY when there has been a collision. With full-duplex links, this would always be low, so this signal's absence implies either that the link is full-duplex or that collisions are handled internally within the PHY. The signal is only asserted high on simultaneous Rx and Tx on half-duplex links. As this must be handled by the PHY if the signal is not available, it can be safely pulled low. |
| FEC CRS  | Carrier sense is asserted high by the PHY to indicate that receive medium is non-idle. As in the case with the COL signal, this never occurs with full-duplex links. Again, it is recommended that this signal be pulled low using an external pulldown.                                                                                                                                                                              |
| FEC_RXER | This signal is asserted high by the PHY to signal an error to the MAC. However, in these circumstances the PHY will deliberately corrupt the CRC of the incoming frame before passing to the MAC. This input to the FEC can therefore be pulled low and any RXER can be detected by a CRC error in the received frame where the CR bit is set in the receive buffer descriptor (RxBd).                                                |
| FEC_TXER | This FEC output can be left unconfigured and unused. The signal was originally included for the test case in some PHY designs where the MAC layer in the FEC can demand that the PHY corrupt data during transmission. However, it is readily possible to do this within the MAC by setting the ABC (Append Bad CRC) bit in the transmit buffer descriptor (TxBD) for the frame being sent.                                           |

### 40.3.2.3 10 Mbps and 100 Mbps RMII interface

The reduced media independent interface (RMII) is a low-cost alternative to the IEEE802.3 MII standard. This interface provides the functionality of the MII interface on a total of 8 pins instead of 18. The RMII interface for 10/100 Ethernet MAC-PHY interface was defined by an industry consortium and is not currently included in the IEEE 802.3 standard. This functionality is selected as described in the chip configuration chapter, and is reflected in the read-only RCR[RMII\_MODE] bit. The RCR[RMII\_10T] bit determines the speed of operation. The reference clock for RMII is always 50 MHz, but this clock can be divided by 10 within the RCR register to support 10 Mbps operation. The PHY must be configured accordingly.

### 40.3.2.4 10 Mbps 7-wire interface operation

The FEC supports 7-wire interface used by many 10 Mbps Ethernet transceivers. The RCR[MII\_MODE] bit controls this functionality. If this bit is cleared, MII mode is disabled and the 10 Mbps 7-wire mode is enabled.

### 40.3.3 Address recognition options

The address options supported are:

- Promiscuous
- Broadcast reject
- Individual address (hash or exact match)
- Multicast hash match

Address recognition options are discussed in detail in [Ethernet address recognition](#).

### 40.3.4 Internal loopback

Internal loopback mode is selected via RCR[LOOP]. Loopback mode is discussed in detail in [MII internal and external loopback](#).

## 40.4 External signal description

The following table describes the various FEC signals, as well as indicating which signals work in available modes.

**Table 40-3. FEC signal descriptions**

| Signal name | MII            | 7-wire | RMII | Description                                                                                                                                                                                                                                                 |
|-------------|----------------|--------|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| FEC_COL     | X <sup>1</sup> | X      | —    | Asserted upon detection of a collision and remains asserted while the collision persists. This signal is not defined for full-duplex mode.                                                                                                                  |
| FEC_CRG     | X <sup>1</sup> | —      | —    | Carrier sense. When asserted, indicates transmit or receive medium is not idle. In RMII mode, this signal is present on the FEC_RXDV pin.                                                                                                                   |
| FEC_MDC     | X              | —      | X    | Output clock provides a timing reference to the PHY for data transfers on the FEC_MDIO signal.                                                                                                                                                              |
| FEC_MDIO    | X              | —      | X    | Transfers control information between the external PHY and the media-access controller. Data is synchronous to FEC_MDC. This signal is an input after reset. When the FEC operates in 10Mbps 7-wire interface mode, this signal should be connected to VSS. |

*Table continues on the next page...*

**Table 40-3. FEC signal descriptions (continued)**

| Signal name                                                                                                          | MII            | 7-wire | RMII | Description                                                                                                                                                                                                                                                                                                                      |
|----------------------------------------------------------------------------------------------------------------------|----------------|--------|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| FEC_RXCLK                                                                                                            | X              | X      | —    | Provides a timing reference for FEC_RXDV, FEC_RXD[3:0], and FEC_RXER.                                                                                                                                                                                                                                                            |
| FEC_RXDV                                                                                                             | X              | X      | X    | Asserting the FEC_RXDV input indicates PHY has valid nibbles present on the MII. FEC_RXDV must remain asserted from the first recovered nibble of the frame through to the last nibble. Assertion of FEC_RXDV must start no later than the SFD and exclude any EOF.<br><br>In RMII mode, this pin also generates the CRS signal. |
| FEC_RXD0                                                                                                             | X              | X      | X    | This pin contains the Ethernet input data transferred from PHY to the media-access controller when FEC_RXDV is asserted.                                                                                                                                                                                                         |
| FEC_RXD1                                                                                                             | X              | —      | X    | This pin contains the Ethernet input data transferred from PHY to the media access controller when FEC_RXDV is asserted.                                                                                                                                                                                                         |
| FEC_RXD[3:2]                                                                                                         | X              | —      | —    | These pins contain the Ethernet input data transferred from PHY to the media access controller when FEC_RXDV is asserted.                                                                                                                                                                                                        |
| FEC_RXER                                                                                                             | X <sup>1</sup> | —      | X    | When asserted with FEC_RXDV, indicates PHY detects an error in the current frame. When FEC_RXDV is not asserted, FEC_RXER has no effect.                                                                                                                                                                                         |
| FEC_TXCLK                                                                                                            | X              | X      | X    | Input clock which provides a timing reference for FEC_TXEN, FEC_TXD[3:0] and FEC_TXER.<br><br>In RMII mode, this signal is the reference clock for receive, transmit, and the control interface.                                                                                                                                 |
| FEC_TXD0                                                                                                             | X              | X      | X    | The serial output Ethernet data and only valid during the assertion of FEC_TXEN.                                                                                                                                                                                                                                                 |
| FEC_TXD1                                                                                                             | X              | —      | X    | This pin contains the serial output Ethernet data and valid only during assertion of FEC_TXEN.                                                                                                                                                                                                                                   |
| FEC_TXD[3:2]                                                                                                         | X              | —      | —    | These pins contain the serial output Ethernet data and valid only during assertion of FEC_TXEN.                                                                                                                                                                                                                                  |
| FEC_TXEN                                                                                                             | X              | X      | X    | Indicates when valid nibbles are present on the MII. This signal is asserted with the first nibble of a preamble and is negated before the first FEC_TXCLK following the final nibble of the frame.                                                                                                                              |
| FEC_TXER                                                                                                             | X <sup>1</sup> | —      | —    | When asserted for one or more clock cycles while FEC_TXEN is also asserted, PHY sends one or more illegal symbols. FEC_TXER has no effect at 10 Mbps or when FEC_TXEN is negated.                                                                                                                                                |
| Note: Interface selection is chip-specific; see the chapter that describes how modules are configured and connected. |                |        |      |                                                                                                                                                                                                                                                                                                                                  |

1. This signal is not available in the MII-Lite interface.

## 40.5 Memory map and register definition

The FEC is programmed by a combination of control/status registers (CSRs) and buffer descriptors. The CSRs control operation modes and extract global status information. The descriptors pass data buffers and related buffer information between the hardware and software.

Each FEC implementation requires a 1 KB memory map space, which is divided into two sections of 512 bytes each for:

- Control/status registers
- Event/statistic counters held in the MIB block

The following table defines the top-level memory map.

**Table 40-4. FEC memory map**

| Address offset | Function                 |
|----------------|--------------------------|
| 000h–1FFh      | Control/status registers |
| 200h–3FFh      | MIB block counters       |

The following table shows the FEC register map. Any address offset not explicitly mentioned in this table is reserved. Do not modify the content of reserved fields. All accesses to and from the registers must be via 32-bit accesses. There is no support for accesses other than 32-bit. Accessing reserved or unimplemented register spaces in the FEC will not return transfer errors.

The MIB counters (shown in the following table) define the locations in the MIB RAM space where hardware-maintained counters reside. The counters are divided into two groups:

- RMON counters include the Ethernet statistics counters defined in RFC 1757
- Truncated frames counter (only frames with lengths up to 2047 bytes are supported)

The transmit and receive RMON counters are independent, which ensures accurate network statistics when operating in full duplex mode.

The included IEEE counters support the mandatory and recommended counter packages defined in Section 5 of ANSI/IEEE Std. 802.3 (1998 edition). The FEC supports IEEE Basic Package objects, but these do not require counters in the MIB block. In addition, some of the recommended package objects supported do not require MIB counters. Counters for transmit and receive full duplex flow control frames are also included.

**FEC memory map**

| Address offset (hex) | Register name                                  | Width (in bits) | Access | Reset value | Section/ page               |
|----------------------|------------------------------------------------|-----------------|--------|-------------|-----------------------------|
| 4                    | Interrupt Event Register (FEC_EIR)             | 32              | w1c    | 0000_0000h  | <a href="#">40.5.1/1965</a> |
| 8                    | Interrupt Mask Register (FEC_EIMR)             | 32              | R/W    | 0000_0000h  | <a href="#">40.5.2/1967</a> |
| 10                   | Receive Descriptor Active Register (FEC_RDAR)  | 32              | R/W    | 0000_0000h  | <a href="#">40.5.3/1969</a> |
| 14                   | Transmit Descriptor Active Register (FEC_TDAR) | 32              | R/W    | 0000_0000h  | <a href="#">40.5.4/1970</a> |
| 24                   | Ethernet Control Register (FEC_ECR)            | 32              | R/W    | F000_0000h  | <a href="#">40.5.5/1971</a> |
| 40                   | MII Management Frame Register (FEC_MMFR)       | 32              | R/W    | Undefined   | <a href="#">40.5.6/1972</a> |
| 44                   | MII Speed Control Register (FEC_MSCR)          | 32              | R/W    | 0000_0000h  | <a href="#">40.5.7/1973</a> |

*Table continues on the next page...*

## FEC memory map (continued)

| Address offset (hex) | Register name                                                  | Width (in bits) | Access | Reset value                 | Section/ page                |
|----------------------|----------------------------------------------------------------|-----------------|--------|-----------------------------|------------------------------|
| 64                   | MIB Control Register (FEC_MIBC)                                | 32              | R/W    | C000_0000h                  | <a href="#">40.5.8/1974</a>  |
| 84                   | Receive Control Register (FEC_RCR)                             | 32              | R/W    | 05EE_0001h                  | <a href="#">40.5.9/1976</a>  |
| C4                   | Transmit Control Register (FEC_TCR)                            | 32              | R/W    | 0000_0000h                  | <a href="#">40.5.10/1978</a> |
| E4                   | Physical Address Low Register (FEC_PALR)                       | 32              | R/W    | <a href="#">See section</a> | <a href="#">40.5.11/1979</a> |
| E8                   | Physical Address High Register and Type Field (FEC_PAUR)       | 32              | R/W    | <a href="#">See section</a> | <a href="#">40.5.12/1980</a> |
| EC                   | Opcode/Pause Duration (FEC_OPD)                                | 32              | R/W    | <a href="#">See section</a> | <a href="#">40.5.13/1980</a> |
| 118                  | Descriptor Individual Upper Address Register (FEC_IAUR)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">40.5.14/1981</a> |
| 11C                  | Descriptor Individual Lower Address Register (FEC_IALR)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">40.5.15/1981</a> |
| 120                  | Descriptor Group Upper Address Register (FEC_GAUR)             | 32              | R/W    | <a href="#">See section</a> | <a href="#">40.5.16/1982</a> |
| 124                  | Descriptor Group Lower Address Register (FEC_GALR)             | 32              | R/W    | <a href="#">See section</a> | <a href="#">40.5.17/1982</a> |
| 144                  | Transmit FIFO Watermark (FEC_TFWR)                             | 32              | R/W    | 0000_0000h                  | <a href="#">40.5.18/1983</a> |
| 14C                  | FIFO Receive Bound Register (FEC_FRBR)                         | 32              | R      | 0000_0600h                  | <a href="#">40.5.19/1984</a> |
| 150                  | FIFO Receive Start Register (FEC_FRSR)                         | 32              | R/W    | 0000_0500h                  | <a href="#">40.5.20/1984</a> |
| 180                  | Receive Descriptor Ring Start Register (FEC_ERDSR)             | 32              | R/W    | <a href="#">See section</a> | <a href="#">40.5.21/1985</a> |
| 184                  | Transmit Buffer Descriptor Ring Start Register (FEC_ETDSR)     | 32              | R/W    | <a href="#">See section</a> | <a href="#">40.5.22/1986</a> |
| 188                  | Receive Buffer Size Register (FEC_EMRBR)                       | 32              | R/W    | <a href="#">See section</a> | <a href="#">40.5.23/1986</a> |
| 200                  | Count of frames not counted correctly (FEC_RMON_T_DROP)        | 32              | R/W    | Undefined                   | <a href="#">40.5.24/1987</a> |
| 204                  | RMON Tx packet count (FEC_RMON_T_PACKETS)                      | 32              | R/W    | Undefined                   | <a href="#">40.5.25/1988</a> |
| 208                  | RMON Tx broadcast packets (FEC_RMON_T_BC_PKT)                  | 32              | R/W    | Undefined                   | <a href="#">40.5.26/1988</a> |
| 20C                  | RMON Tx multicast packets (FEC_RMON_T_MC_PKT)                  | 32              | R/W    | Undefined                   | <a href="#">40.5.27/1989</a> |
| 210                  | RMON Tx packets with CRC/align error (FEC_RMON_T_CRC_ALIGN)    | 32              | R/W    | Undefined                   | <a href="#">40.5.28/1989</a> |
| 214                  | RMON Tx packets < 64 bytes, good CRC (FEC_RMON_T_UNDERSIZE)    | 32              | R/W    | Undefined                   | <a href="#">40.5.29/1990</a> |
| 218                  | RMON Tx packets > MAX_FL bytes, good CRC (FEC_RMON_T_OVERSIZE) | 32              | R/W    | Undefined                   | <a href="#">40.5.30/1990</a> |

Table continues on the next page...

## FEC memory map (continued)

| Address offset (hex) | Register name                                                       | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|---------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 21C                  | RMON Tx packets < 64 bytes, bad CRC (FEC_RMON_T_FRAG)               | 32              | R/W    | Undefined   | <a href="#">40.5.31/1991</a> |
| 220                  | RMON Tx packets > MAX_FL bytes, bad CRC (FEC_RMON_T_JAB)            | 32              | R/W    | Undefined   | <a href="#">40.5.32/1991</a> |
| 224                  | RMON Tx collision count (FEC_RMON_T_COL)                            | 32              | R/W    | Undefined   | <a href="#">40.5.33/1992</a> |
| 228                  | RMON Tx 64 byte packets (FEC_RMON_T_P64)                            | 32              | R/W    | Undefined   | <a href="#">40.5.34/1992</a> |
| 22C                  | RMON Tx 65 to 127 byte packets (FEC_RMON_T_P65TO127)                | 32              | R/W    | Undefined   | <a href="#">40.5.35/1993</a> |
| 230                  | RMON Tx 128 to 255 byte packets (FEC_RMON_T_P128TO255)              | 32              | R/W    | Undefined   | <a href="#">40.5.36/1993</a> |
| 234                  | RMON Tx 256 to 511 byte packets (FEC_RMON_T_P256TO511)              | 32              | R/W    | Undefined   | <a href="#">40.5.37/1994</a> |
| 238                  | RMON Tx 512 to 1023 byte packets (FEC_RMON_T_P512TO1023)            | 32              | R/W    | Undefined   | <a href="#">40.5.38/1994</a> |
| 23C                  | RMON Tx 1024 to 2047 byte packets (FEC_RMON_T_P1024TO2047)          | 32              | R/W    | Undefined   | <a href="#">40.5.39/1995</a> |
| 240                  | RMON Tx packets with > 2048 bytes (FEC_RMON_T_P_GTE2048)            | 32              | R/W    | Undefined   | <a href="#">40.5.40/1995</a> |
| 244                  | RMON Tx Octets (FEC_RMON_T_OCTETS)                                  | 32              | R/W    | Undefined   | <a href="#">40.5.41/1996</a> |
| 248                  | Count of transmitted frames not counted correctly (FEC_IEEE_T_DROP) | 32              | R/W    | Undefined   | <a href="#">40.5.42/1996</a> |
| 24C                  | Frames transmitted OK (FEC_IEEE_T_FRAME_OK)                         | 32              | R/W    | Undefined   | <a href="#">40.5.43/1997</a> |
| 250                  | Frames transmitted with single collision (FEC_IEEE_T_1COL)          | 32              | R/W    | Undefined   | <a href="#">40.5.44/1997</a> |
| 254                  | Frames transmitted with multiple collisions (FEC_IEEE_T_MCOL)       | 32              | R/W    | Undefined   | <a href="#">40.5.45/1998</a> |
| 258                  | Frames transmitted after deferral delay (FEC_IEEE_T_DEF)            | 32              | R/W    | Undefined   | <a href="#">40.5.46/1998</a> |
| 25C                  | Frames transmitted with late collision (FEC_IEEE_T_LCOL)            | 32              | R/W    | Undefined   | <a href="#">40.5.47/1999</a> |
| 260                  | Frames transmitted with excessive collisions (FEC_IEEE_T_EXCOL)     | 32              | R/W    | Undefined   | <a href="#">40.5.48/1999</a> |
| 264                  | Frames transmitted with Tx FIFO underrun (FEC_IEEE_T_MACERR)        | 32              | R/W    | Undefined   | <a href="#">40.5.49/2000</a> |
| 268                  | Frames transmitted with carrier sense error (FEC_IEEE_T_CSERR)      | 32              | R/W    | Undefined   | <a href="#">40.5.50/2000</a> |
| 26C                  | Frames transmitted with SQE error (FEC_IEEE_T_SQE)                  | 32              | R/W    | Undefined   | <a href="#">40.5.51/2001</a> |
| 270                  | Flow control pause frames transmitted (FEC_IEEE_T_FDXFC)            | 32              | R/W    | Undefined   | <a href="#">40.5.52/2001</a> |

Table continues on the next page...

## FEC memory map (continued)

| Address offset (hex) | Register name                                                           | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|-------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 274                  | Octet count for frames transmitted without error (FEC_IEEE_T_OCTETS_OK) | 32              | R/W    | Undefined   | <a href="#">40.5.53/2002</a> |
| 280                  | Count of received frames not counted correctly (FEC_RMON_R_DROP)        | 32              | R/W    | Undefined   | <a href="#">40.5.54/2002</a> |
| 284                  | RMON Rx packet count (FEC_RMON_R_PACKETS)                               | 32              | R/W    | Undefined   | <a href="#">40.5.55/2003</a> |
| 288                  | RMON Rx broadcast packets (FEC_RMON_R_BC_PKT)                           | 32              | R/W    | Undefined   | <a href="#">40.5.56/2003</a> |
| 28C                  | RMON Rx multicast packets (FEC_RMON_R_MC_PKT)                           | 32              | R/W    | Undefined   | <a href="#">40.5.57/2004</a> |
| 290                  | RMON Rx packets with CRC/Align error (FEC_RMON_R_CRC_ALIGN)             | 32              | R/W    | Undefined   | <a href="#">40.5.58/2004</a> |
| 294                  | RMON Rx packets < 64 bytes, good CRC (FEC_RMON_R_UNDERSIZE)             | 32              | R/W    | Undefined   | <a href="#">40.5.59/2005</a> |
| 298                  | RMON Rx packets > MAX_FL bytes, good CRC (FEC_RMON_R_OVERSIZE)          | 32              | R/W    | Undefined   | <a href="#">40.5.60/2005</a> |
| 29C                  | RMON Rx packets < 64 bytes, bad CRC (FEC_RMON_R_FRAG)                   | 32              | R/W    | Undefined   | <a href="#">40.5.61/2006</a> |
| 2A0                  | RMON Rx packets > MAX_FL bytes, bad CRC (FEC_RMON_R_JAB)                | 32              | R/W    | Undefined   | <a href="#">40.5.62/2006</a> |
| 2A4                  | Reserved (FEC_RMON_R_RESVD_0)                                           | 32              | R/W    | Undefined   | <a href="#">40.5.63/2007</a> |
| 2A8                  | RMON Rx 64 byte packets (FEC_RMON_R_P64)                                | 32              | R/W    | Undefined   | <a href="#">40.5.64/2007</a> |
| 2AC                  | RMON Rx 65 to 127 byte packets (FEC_RMON_R_P65TO127)                    | 32              | R/W    | Undefined   | <a href="#">40.5.65/2008</a> |
| 2B0                  | RMON Rx 128 to 255 byte packets (FEC_RMON_R_P128TO255)                  | 32              | R/W    | Undefined   | <a href="#">40.5.66/2008</a> |
| 2B4                  | RMON Rx 256 to 511 byte packets (FEC_RMON_R_P256TO511)                  | 32              | R/W    | Undefined   | <a href="#">40.5.67/2009</a> |
| 2B8                  | RMON Rx 512 to 1023 byte packets (FEC_RMON_R_P512TO1023)                | 32              | R/W    | Undefined   | <a href="#">40.5.68/2009</a> |
| 2BC                  | RMON Rx 1024 to 2047 byte packets (FEC_RMON_R_P1024TO2047)              | 32              | R/W    | Undefined   | <a href="#">40.5.69/2010</a> |
| 2C0                  | RMON Rx packets with > 2048 bytes (FEC_RMON_R_P_GTE2048)                | 32              | R/W    | Undefined   | <a href="#">40.5.70/2010</a> |
| 2C4                  | RMON Rx octets (FEC_RMON_R_OCTETS)                                      | 32              | R/W    | Undefined   | <a href="#">40.5.71/2011</a> |
| 2C8                  | Count of received frames not counted correctly (FEC_IEEE_R_DROP)        | 32              | R/W    | Undefined   | <a href="#">40.5.72/2011</a> |
| 2CC                  | Frames received OK (FEC_IEEE_R_FRAME_OK)                                | 32              | R/W    | Undefined   | <a href="#">40.5.73/2012</a> |
| 2D0                  | Frames received with CRC error (FEC_IEEE_R_CRC)                         | 32              | R/W    | Undefined   | <a href="#">40.5.74/2012</a> |

Table continues on the next page...



**FEC memory map (continued)**

| Address offset (hex) | Register name                                                        | Width (in bits) | Access | Reset value | Section/page                 |
|----------------------|----------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 2D4                  | Frames received with alignment error (FEC_IEEE_R_ALIGN)              | 32              | R/W    | Undefined   | <a href="#">40.5.75/2013</a> |
| 2D8                  | Receive FIFO overflow count (FEC_IEEE_R_MACERR)                      | 32              | R/W    | Undefined   | <a href="#">40.5.76/2013</a> |
| 2DC                  | Flow control pause frames received (FEC_IEEE_R_FDXFC)                | 32              | R/W    | Undefined   | <a href="#">40.5.77/2014</a> |
| 2E0                  | Octet count for frames received without error (FEC_IEEE_R_OCTETS_OK) | 32              | R/W    | Undefined   | <a href="#">40.5.78/2014</a> |

**40.5.1 Interrupt Event Register (FEC\_EIR)**

When an event occurs that sets a bit in EIR, an interrupt occurs if the corresponding bit in the interrupt mask register (EIMR) is also set. Writing a 1 to an EIR bit clears it; writing 0 has no effect. This register is cleared upon hardware reset.

These interrupts can be divided into operational interrupts, transceiver/network error interrupts, and internal error interrupts. Interrupts which may occur in normal operation are GRA, TXF, TXB, RXF, RXB, and MII. Interrupts resulting from errors/problems detected in the network or transceiver are HBERR, BABR, BABT, LC, and RL. Interrupts resulting from internal errors are HBERR and UN.

Some of the error interrupts are independently counted in the MIB block counters:

- HBERR - IEEE\_T\_SQE
- BABR - RMON\_R\_OVERSIZE (good CRC), RMON\_R\_JAB (bad CRC)
- BABT - RMON\_T\_OVERSIZE (good CRC), RMON\_T\_JAB (bad CRC)
- LATE\_COL - IEEE\_T\_LCOL
- COL\_RETRY\_LIM - IEEE\_T\_EXCOL
- XFIFO\_UN - IEEE\_T\_MACERR

Software may choose to mask off these interrupts because these errors are visible to network management via the MIB counters.

## Memory map and register definition

Address: 0h base + 4h offset = 4h

|       |       |      |      |     |     |     |     |     |     |       |     |     |     |    |    |    |
|-------|-------|------|------|-----|-----|-----|-----|-----|-----|-------|-----|-----|-----|----|----|----|
| Bit   | 0     | 1    | 2    | 3   | 4   | 5   | 6   | 7   | 8   | 9     | 10  | 11  | 12  | 13 | 14 | 15 |
| R     | HBERR | BABR | BABT | GRA | TXF | TXB | RXF | RXB | MII | EBERR | LC  | RL  | UN  | 0  |    |    |
| W     | w1c   | w1c  | w1c  | w1c | w1c | w1c | w1c | w1c | w1c | w1c   | w1c | w1c | w1c |    |    |    |
| Reset | 0     | 0    | 0    | 0   | 0   | 0   | 0   | 0   | 0   | 0     | 0   | 0   | 0   | 0  | 0  | 0  |
| Bit   | 16    | 17   | 18   | 19  | 20  | 21  | 22  | 23  | 24  | 25    | 26  | 27  | 28  | 29 | 30 | 31 |
| R     | 0     |      |      |     |     |     |     |     |     |       |     |     |     |    |    |    |
| W     |       |      |      |     |     |     |     |     |     |       |     |     |     |    |    |    |
| Reset | 0     | 0    | 0    | 0   | 0   | 0   | 0   | 0   | 0   | 0     | 0   | 0   | 0   | 0  | 0  | 0  |

### FEC\_EIR field descriptions

| Field      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>HBERR | Heartbeat error. Indicates TCR[HBC] is set and that the COL input was not asserted within the heartbeat window following a transmission.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 1<br>BABR  | Babbling receive error. Indicates a frame was received with length in excess of RCR[MAX_FL] bytes.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 2<br>BABT  | Babbling transmit error. Indicates the transmitted frame length exceeds RCR[MAX_FL] bytes. Usually this condition is caused by a frame that is too long is placed into the transmit data buffer(s). Truncation does not occur.                                                                                                                                                                                                                                                                                                                                                                                                            |
| 3<br>GRA   | Graceful stop complete. Indicates the graceful stop is complete. During graceful stop the transmitter is placed into a pause state after completion of the frame currently being transmitted. This bit is set by one of these conditions: <ul style="list-style-type: none"> <li>A graceful stop initiated by the setting of the TCR[GTS] bit is now complete.</li> <li>A graceful stop initiated by the setting of the TCR[TFC_PAUSE] bit is now complete.</li> <li>A graceful stop initiated by the reception of a valid full duplex flow control pause frame is now complete. See <a href="#">Full duplex flow control</a>.</li> </ul> |
| 4<br>TXF   | Transmit frame interrupt. Indicates a frame has been transmitted and the last corresponding buffer descriptor has been updated.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 5<br>TXB   | Transmit buffer interrupt. Indicates a transmit buffer descriptor has been updated.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 6<br>RXF   | Receive frame interrupt. Indicates a frame has been received and the last corresponding buffer descriptor has been updated.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 7<br>RXB   | Receive buffer interrupt. Indicates a receive buffer descriptor not the last in the frame has been updated.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 8<br>MII   | MII interrupt. Indicates the MII has completed the data transfer requested.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 9<br>EBERR | Ethernet bus error. Indicates a system bus error occurred when a DMA transaction is underway. When the EBERR bit is set, ECR[ETHER_EN] is cleared, halting frame processing by the FEC. When this occurs, software needs to ensure that the FIFO controller and DMA also soft reset.                                                                                                                                                                                                                                                                                                                                                      |

Table continues on the next page...

**FEC\_EIR field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                  |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 10<br>LC          | Late collision. Indicates a collision occurred beyond the collision window (slot time) in half duplex mode. The frame truncates with a bad CRC and the remainder of the frame is discarded.                                                                  |
| 11<br>RL          | Collision retry limit. Indicates a collision occurred on each of 16 successive attempts to transmit the frame. The frame is discarded without being transmitted and transmission of the next frame commences. This error can only occur in half duplex mode. |
| 12<br>UN          | Transmit FIFO underrun. Indicates the transmit FIFO became empty before the complete frame was transmitted. A bad CRC is appended to the frame fragment and the remainder of the frame is discarded.                                                         |
| 13–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                      |

**40.5.2 Interrupt Mask Register (FEC\_EIMR)**

The EIMR controls which interrupt events are allowed to generate actual interrupts. A hardware reset clears this register. If the corresponding bits in the EIR and EIMR registers are set, an interrupt is generated. The interrupt signal remains asserted until a 1 is written to the EIR bit (write 1 to clear) or a 0 is written to the EIMR bit.

Each bit in this register corresponds to an interrupt source defined by the EIR. The corresponding EIMR bit determines whether an interrupt condition can generate an interrupt. At every processor clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR bit reflects the state of the interrupt signal even if the corresponding EIMR bit is set.

Address: 0h base + 8h offset = 8h

|       |       |      |      |     |     |     |     |     |     |       |    |    |    |    |    |    |
|-------|-------|------|------|-----|-----|-----|-----|-----|-----|-------|----|----|----|----|----|----|
| Bit   | 0     | 1    | 2    | 3   | 4   | 5   | 6   | 7   | 8   | 9     | 10 | 11 | 12 | 13 | 14 | 15 |
| R     |       |      |      |     |     |     |     |     |     |       |    |    |    |    |    |    |
| W     | HBERR | BABR | BABT | GRA | TXF | TXB | RXF | RXB | MII | EBERR | LC | RL | UN |    |    |    |
| Reset | 0     | 0    | 0    | 0   | 0   | 0   | 0   | 0   | 0   | 0     | 0  | 0  | 0  | 0  | 0  | 0  |
| Bit   | 16    | 17   | 18   | 19  | 20  | 21  | 22  | 23  | 24  | 25    | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |       |      |      |     |     |     |     |     |     |       |    |    |    |    |    |    |
| W     |       |      |      |     |     |     |     |     |     |       |    |    |    |    |    |    |
| Reset | 0     | 0    | 0    | 0   | 0   | 0   | 0   | 0   | 0   | 0     | 0  | 0  | 0  | 0  | 0  | 0  |

**FEC\_EIMR field descriptions**

| Field      | Description                     |
|------------|---------------------------------|
| 0<br>HBERR | Heartbeat error interrupt mask. |

*Table continues on the next page...*

**FEC\_EIMR field descriptions (continued)**

| Field      | Description                                                                                                               |
|------------|---------------------------------------------------------------------------------------------------------------------------|
|            | 0 The interrupt source is masked.<br>1 The interrupt source is not masked.                                                |
| 1<br>BABR  | Babbling receive error interrupt mask.<br><br>0 The interrupt source is masked.<br>1 The interrupt source is not masked.  |
| 2<br>BABT  | Babbling transmit error interrupt mask.<br><br>0 The interrupt source is masked.<br>1 The interrupt source is not masked. |
| 3<br>GRA   | Graceful stop complete interrupt mask.<br><br>0 The interrupt source is masked.<br>1 The interrupt source is not masked.  |
| 4<br>TXF   | Transmit frame interrupt mask.<br><br>0 The interrupt source is masked.<br>1 The interrupt source is not masked.          |
| 5<br>TXB   | Transmit buffer interrupt mask.<br><br>0 The interrupt source is masked.<br>1 The interrupt source is not masked.         |
| 6<br>RXF   | Receive frame interrupt mask.<br><br>0 The interrupt source is masked.<br>1 The interrupt source is not masked.           |
| 7<br>RXB   | Receive buffer interrupt mask.<br><br>0 The interrupt source is masked.<br>1 The interrupt source is not masked.          |
| 8<br>MII   | MII interrupt mask.<br><br>0 The interrupt source is masked.<br>1 The interrupt source is not masked.                     |
| 9<br>EBERR | Ethernet bus error interrupt mask.<br><br>0 The interrupt source is masked.<br>1 The interrupt source is not masked.      |
| 10<br>LC   | Late collision interrupt mask.<br><br>0 The interrupt source is masked.<br>1 The interrupt source is not masked.          |
| 11<br>RL   | Collision retry limit interrupt mask.<br><br>0 The interrupt source is masked.<br>1 The interrupt source is not masked.   |
| 12<br>UN   | Transmit FIFO underrun interrupt mask.                                                                                    |

*Table continues on the next page...*

**FEC\_EIMR field descriptions (continued)**

| Field             | Description                                                                             |
|-------------------|-----------------------------------------------------------------------------------------|
|                   | 0 The interrupt source is masked.<br>1 The interrupt source is not masked.              |
| 13–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |

**40.5.3 Receive Descriptor Active Register (FEC\_RDAR)**

RDAR is a command register, written by the user, indicating the receive descriptor ring is updated (the driver produced empty receive buffers with the empty bit set).

When the register is written, the RDAR bit is set. This is independent of the data actually written by the user. When set, the FEC polls the receive descriptor ring and processes receive frames (if ECR[ETHER\_EN] is also set). After the FEC polls a receive descriptor whose empty bit is not set, FEC clears the RDAR bit and ceases receive descriptor ring polling until the user sets the bit again, signifying that additional descriptors are placed into the receive descriptor ring.

The RDAR is cleared at reset and when ECR[ETHER\_EN] is cleared.

Address: 0h base + 10h offset = 10h

|       |    |    |    |    |    |    |    |      |    |    |    |    |    |    |    |    |
|-------|----|----|----|----|----|----|----|------|----|----|----|----|----|----|----|----|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7    | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0  |    |    |    |    |    |    | RDAR | 0  |    |    |    |    |    |    |    |
| W     |    |    |    |    |    |    |    |      |    |    |    |    |    |    |    |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23   | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0  |    |    |    |    |    |    |      |    |    |    |    |    |    |    |    |
| W     |    |    |    |    |    |    |    |      |    |    |    |    |    |    |    |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**FEC\_RDAR field descriptions**

| Field            | Description                                                                                                                                                                                              |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–6<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                  |
| 7<br>RDAR        | Set to 1 when this register is written, regardless of the value written. Cleared by the FEC when no additional empty descriptors remain in the receive ring. Also cleared when ECR[ETHER_EN] is cleared. |
| 8–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                  |

40.5.4 Transmit Descriptor Active Register (FEC\_TDAR)

The TDAR is a command register which the user writes to indicate the transmit descriptor ring is updated (transmit buffers have been produced by the driver with the ready bit set in the buffer descriptor).

When the register is written, the TDAR field is set. This value is independent of the data actually written by the user. When set, the FEC polls the transmit descriptor ring and processes transmit frames (provided ECR[ETHER\_EN] is also set). After the FEC polls a transmit descriptor that is a ready bit not set, FEC clears the TDAR field and ceases transmit descriptor ring polling until the user sets the bit again, signifying additional descriptors are placed into the transmit descriptor ring.

The TDAR field is cleared when any of the following events occur:

- Chip reset
- When ECR[ETHER\_EN] is cleared
- When ECR[RESET] is set

Address: 0h base + 14h offset = 14h

|       |   |   |   |   |   |   |   |      |   |   |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|------|---|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7    | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0 |   |   |   |   |   |   | TDAR | 0 |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |      |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0    | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

FEC\_TDAR field descriptions

| Field            | Description                                                                                                                                                                                                      |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–6<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                          |
| 7<br>TDAR        | Set to 1 when this register is written, regardless of the value written. Cleared by the FEC device when no additional ready descriptors remain in the transmit ring. Also cleared when ECR[ETHER_EN] is cleared. |
| 8–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                          |

## 40.5.5 Ethernet Control Register (FEC\_ECR)

ECR is a read/write user register, though hardware may alter fields in this register as well. The ECR enables/disables the FEC.

Address: 0h base + 24h offset = 24h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0 |   |   |   | 0 | 0 | 0 |   |   |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |       |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|-------|
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30       | 31    |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |    | 0  | ETHER_EN | RESET |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |       |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0        | 0     |

### FEC\_ECR field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–3<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 4<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 5<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 6–28<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 29<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 30<br>ETHER_EN   | When this bit is set, FEC is enabled, and reception and transmission are possible. When this bit is cleared, reception immediately stops and transmission stops after a bad CRC is appended to any currently transmitted frame. The buffer descriptor(s) for an aborted transmit frame are not updated after clearing this bit. When ETHER_EN is cleared, the DMA, buffer descriptor, and FIFO control logic are reset, including the buffer descriptor and FIFO pointers. Hardware alters the ETHER_EN bit under the following conditions: <ul style="list-style-type: none"> <li>• ECR[RESET] is set by software, in which case ETHER_EN is cleared.</li> <li>• An error condition causes the EIR[EBERR] bit to set, in which case ETHER_EN is cleared.</li> </ul> |
| 31<br>RESET      | When this bit is set, the equivalent of a hardware reset is performed but it is local to the FEC. ECR[ETHER_EN] is cleared and all other FEC registers take their reset values. Also, any transmission/reception currently in progress is abruptly aborted. This bit is automatically cleared by hardware during the reset sequence. The reset sequence takes approximately eight internal bus clock cycles after this bit is set.                                                                                                                                                                                                                                                                                                                                   |

### 40.5.6 MII Management Frame Register (FEC\_MMFR)

The MMFR is used to communicate with the attached MII compatible PHY device(s), providing read/write access to their MII registers. Performing a write to the MMFR causes a management frame to be sourced unless the MSCR is programmed to 0. If MSCR is cleared while MMFR is written and then MSCR is written with a non-zero value, an MII frame is generated with the data previously written to the MMFR. This allows MMFR and MSCR to be programmed in either order if MSCR is currently zero.

The MMFR does not reset to any predefined value.

To perform a read or write operation on the MII Management Interface, write the MMFR. To generate a valid read or write management frame, ST field must be written with a 01 pattern, and the TA field must be written with a 10. If other patterns are written to these fields, a frame is generated, but does not comply with the IEEE 802.3 MII definition.

To generate an IEEE 802.3-compliant MII Management Interface write frame (write to a PHY register), write {01 01 PHYAD REGAD 10 DATA} to the MMFR. Writing this pattern causes the control logic to shift out the data in the MMFR following a preamble generated by the control state machine. During this time, contents of the MMFR register are altered as the contents are serially shifted and are unpredictable if read by the user. After the write management frame operation completes, the MII interrupt is generated. At this time, contents of the MMFR match the original value written.

To generate an MII management interface read frame (read a PHY register), write {01 10 PHYAD REGAD 10 XXXX} to the MMFR (the content of the DATA field is unimportant). Writing this pattern causes the control logic to shift out the data in the MMFR following a preamble generated by the control state machine. During this time, contents of the MMFR register are altered as the contents are serially shifted and are unpredictable if read by the user. After the read management frame operation completes, the MII interrupt is generated. At this time, the contents of the MMFR register match the original value written except for the DATA field whose contents are replaced by the value read from the PHY register.

If the MMFR register is written while frame generation is in progress, the frame contents are altered. Software must use the MII interrupt to avoid writing to the MMFR while frame generation is in progress.

Address: 0h base + 40h offset = 40h

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     | ST | OP |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* |

\* Notes:



- x = Undefined at reset.

### FEC\_MMFR field descriptions

| Field         | Description                                                                                                                                                                                                                                       |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–1<br>ST     | Start of frame delimiter. These bits must be programmed to 0b01 for a valid MII management frame.                                                                                                                                                 |
| 2–3<br>OP     | Operation code.<br>00 Write frame operation, but not MII compliant.<br>01 Write frame operation for a valid MII management frame.<br>10 Read frame operation for a valid MII management frame.<br>11 Read frame operation, but not MII compliant. |
| 4–8<br>PA     | PHY address. This field specifies one of up to 32 attached PHY devices.                                                                                                                                                                           |
| 9–13<br>RA    | Register address. This field specifies one of up to 32 registers within the specified PHY device.                                                                                                                                                 |
| 14–15<br>TA   | Turn around. This field must be programmed to 10 to generate a valid MII management frame.                                                                                                                                                        |
| 16–31<br>DATA | Management frame data. This is the field for data to be written to or read from the PHY register.                                                                                                                                                 |

## 40.5.7 MII Speed Control Register (FEC\_MSCR)

The MSCR:

- Provides control of the MII clock (FEC\_MDC pin) frequency
- Allows a preamble drop on the MII management frame

The MII\_SPEED field must be programmed with a value to provide an FEC\_MDC frequency of less than or equal to 2.5 MHz to be compliant with the IEEE 802.3 MII specification. The MII\_SPEED must be set to a non-zero value to source a read or write management frame. After the management frame is complete, the MSCR register may optionally be set to 0 to turn off the FEC\_MDC. The FEC\_MDC generated has a 50% duty cycle except when MII\_SPEED changes during operation (change takes effect following a rising or falling edge of FEC\_MDC).

If the internal bus clock is 25 MHz, programming this register to 0000\_0005h results in an FEC\_MDC as stated the equation below.

$$25 \text{ MHz} \times \frac{1}{52} = 2.5 \text{ MHz}$$

**Figure 40-9. FEC\_MDC example**

[Table 40-13](#) shows optimum values for MII\_SPEED as a function of internal bus clock frequency.

**Table 40-13. Programming examples for MSCR**

| Internal FEC clock frequency | MSCR[MII_SPEED] | FEC_MDC frequency |
|------------------------------|-----------------|-------------------|
| 25 MHz                       | 5h              | 2.50 MHz          |
| 33 MHz                       | 7h              | 2.36 MHz          |
| 40 MHz                       | 8h              | 2.50 MHz          |
| 50 MHz                       | Ah              | 2.50 MHz          |
| 66 MHz                       | Eh              | 2.36 MHz          |
| 80 MHz                       | 10h             | 2.50 MHz          |
| 83 MHz                       | 10h             | 2.59 MHz          |

Address: 0h base + 44h offset = 44h

|       |    |    |    |    |    |    |    |    |          |           |    |    |    |    |    |    |  |  |  |          |
|-------|----|----|----|----|----|----|----|----|----------|-----------|----|----|----|----|----|----|--|--|--|----------|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8        | 9         | 10 | 11 | 12 | 13 | 14 | 15 |  |  |  |          |
| R     | 0  |    |    |    |    |    |    |    |          |           |    |    |    |    |    |    |  |  |  |          |
| W     |    |    |    |    |    |    |    |    |          |           |    |    |    |    |    |    |  |  |  |          |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0        | 0         | 0  | 0  | 0  | 0  | 0  | 0  |  |  |  |          |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24       | 25        | 26 | 27 | 28 | 29 | 30 | 31 |  |  |  |          |
| R     | 0  |    |    |    |    |    |    |    | DIS_ PRE | MII_SPEED |    |    |    |    |    |    |  |  |  | Reserved |
| W     |    |    |    |    |    |    |    |    |          |           |    |    |    |    |    |    |  |  |  |          |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0        | 0         | 0  | 0  | 0  | 0  | 0  | 0  |  |  |  |          |

**FEC\_MSCR field descriptions**

| Field              | Description                                                                                                                                                                                                                                                                                                            |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–23<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                |
| 24<br>DIS_PRE      | Setting this bit causes the preamble (32 ones) not to be prepended to the MII management frame. The MII standard allows the preamble to be dropped if the attached PHY device(s) does not require it.                                                                                                                  |
| 25–30<br>MII_SPEED | Controls the frequency of the MII management interface clock (FEC_MDC) relative to the internal bus clock. A value of 0 in this field turns off the FEC_MDC and leaves it in low voltage state. Any non-zero value results in the FEC_MDC frequency of $1/(\text{MII\_SPEED} \times 2)$ of the internal bus frequency. |
| 31<br>Reserved     | This field is reserved.<br>Reserved                                                                                                                                                                                                                                                                                    |

**40.5.8 MIB Control Register (FEC\_MIBC)**

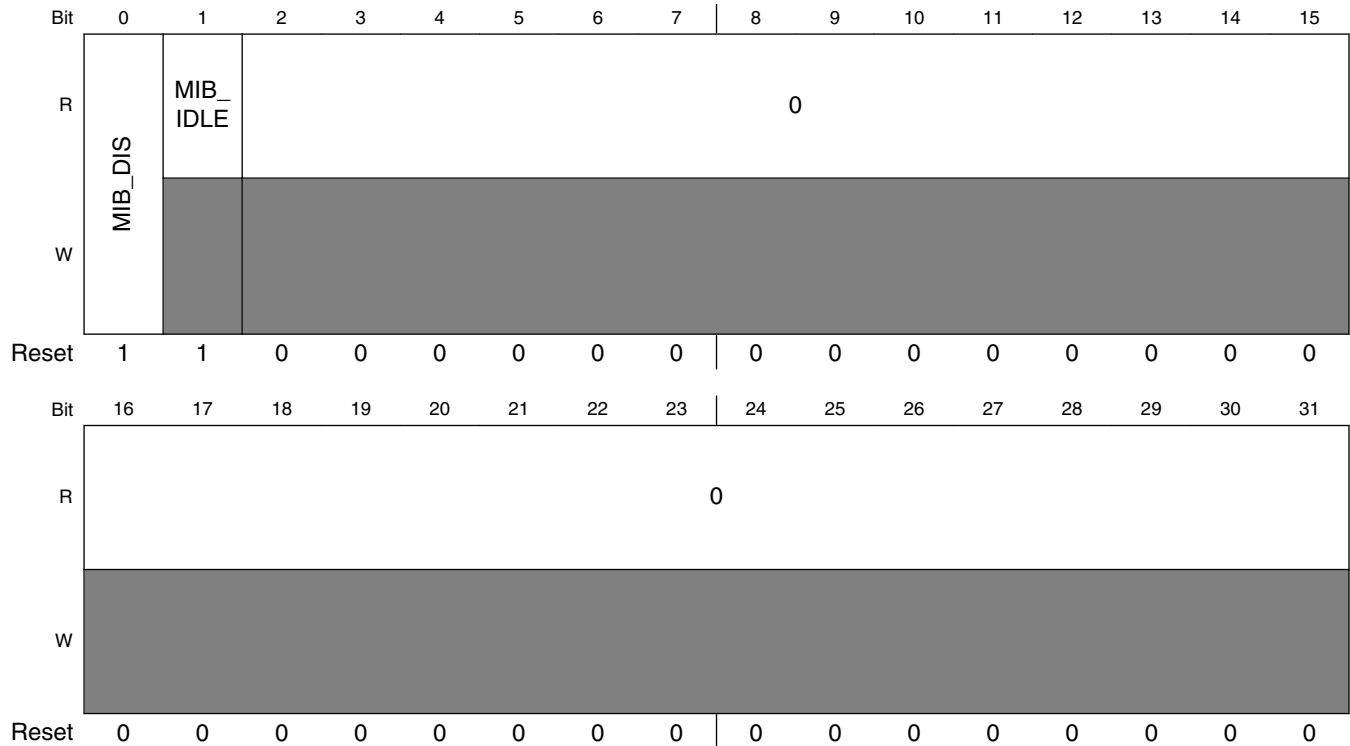
The MIBC is a read/write register controlling and observing the state of the MIB block. User software accesses this register if there is a need to disable the MIB block operation. For example, to clear all MIB counters in RAM:

1. Disable the MIB block

2. Clear all the MIB RAM locations
3. Enable the MIB block

The MIB\_DIS bit is reset to 1. See the memory map for the locations of the MIB counters.

Address: 0h base + 64h offset = 64h



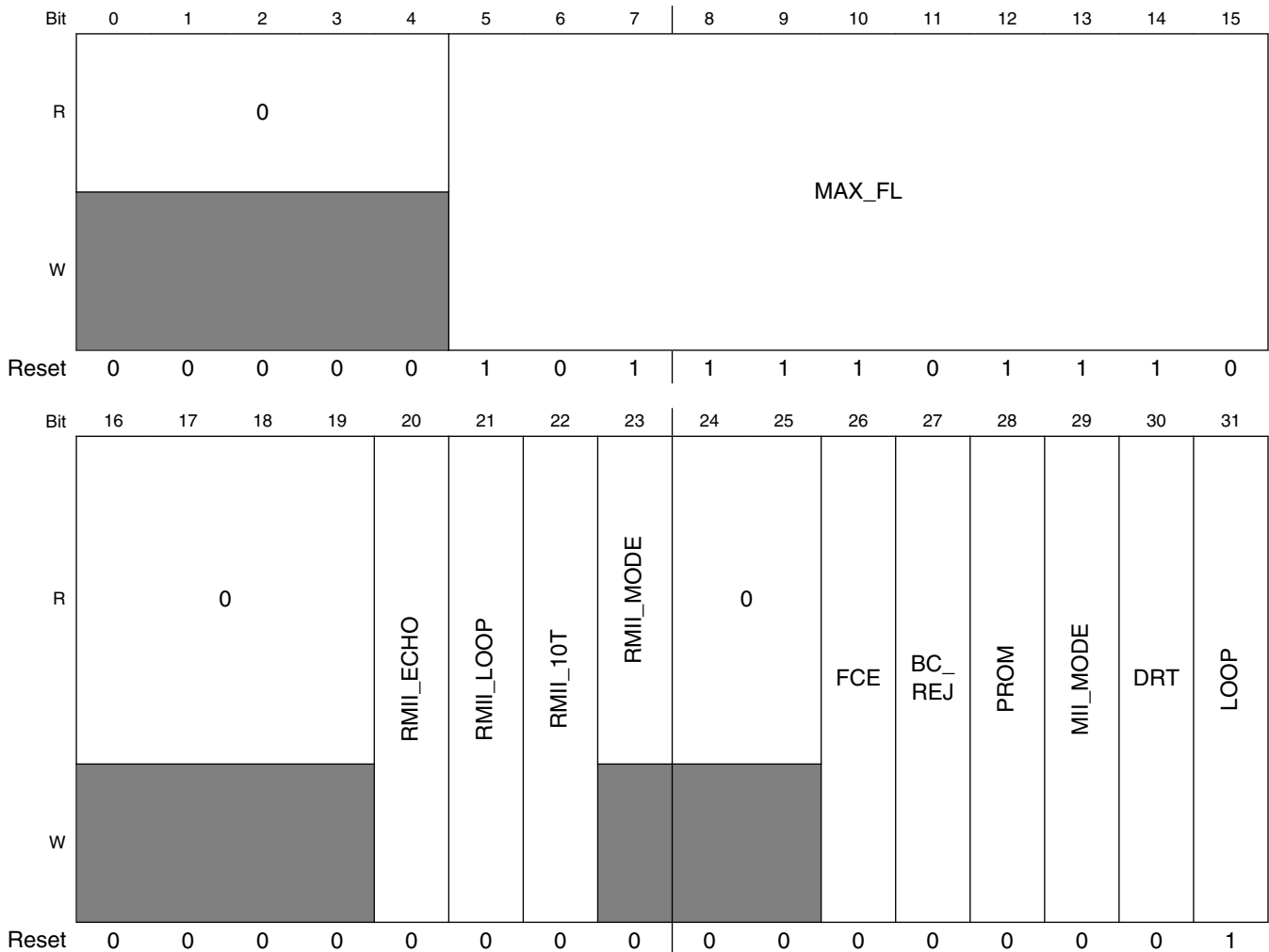
### FEC\_MIBC field descriptions

| Field            | Description                                                                             |
|------------------|-----------------------------------------------------------------------------------------|
| 0<br>MIB_DIS     | If MIB_DIS = 1, the MIB logic halts and not update any MIB counters.                    |
| 1<br>MIB_IDLE    | If MIB_IDLE = 1, the MIB block is not currently updating any MIB counters.              |
| 2–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |

40.5.9 Receive Control Register (FEC\_RCR)

RCR controls the operational mode of the receive block and must be written only when ECR[ETHER\_EN] is cleared (initialization time).

Address: 0h base + 84h offset = 84h



FEC\_RCR field descriptions

| Field           | Description                                                                                                                                                                                                                                                                                                          |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–4<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                              |
| 5–15<br>MAX_FL  | Maximum frame length. Length is measured starting at DA and includes the CRC at the end of the frame. Transmit frames longer than MAX_FL causes the BABT interrupt to occur. Receive frames longer than MAX_FL causes the BABR interrupt to occur and sets the LG bit in the end of frame receive buffer descriptor. |

Table continues on the next page...

## FEC\_RCR field descriptions (continued)

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | The recommended default value to be programmed is 1518 or 1522 if VLAN tags are supported.                                                                                                                                                                                                                                                                                                                                                                                                      |
| 16–19<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                         |
| 20<br>RMII_ECHO   | RMII Echo. Enables RMII echo mode. RMII 2-bit receive data is processed through the RMII receive logic to the FEC and also routes through the RMII's transmit logic to become RMII 2-bit transmit data out.<br><br>0 Normal operation.<br>1 RMII echo mode enabled.<br><br><b>NOTE:</b> When RMII_ECHO is set, proper operation is guaranteed only when RMII_MODE = 1, RMII_LOOP = 0 and LOOP = 0.                                                                                              |
| 21<br>RMII_LOOP   | RMII loopback. Enables RMII loopback mode. Causes the MII transmit outputs from the Ethernet controller to loop back to the Ethernet controllers's MII receive inputs through the RMII transmit/receive logic.<br><br>0 Normal operation.<br>1 RMII loopback mode enabled.<br><br><b>NOTE:</b> When RMII_LOOP is set, proper operation is guaranteed only when RMII_MODE = 1, RMII_ECHO = 0, LOOP = 0, and TCR <i>n</i> [FDEN] = 1.                                                             |
| 22<br>RMII_10T    | RMII 10-Base T. Enables 10Mbps mode of the RMII. Determines the clock frequency of the clock source to the FEC logic to support 10/100Mbps operations.<br><br>0 100 Mbps operation. The 50 MHz RMII reference clock on FEC_TXCLK is sent to the RMII, while a divided-by-2 version (25 MHz) is sent to the FEC.<br>1 10 Mbps operation. The 50 MHz RMII reference clock on FEC_TXCLK is divided by 10 (5 MHz) and sent to the RMII, while a divided-by-20 version (2.5 MHz) is sent to the FEC. |
| 23<br>RMII_MODE   | RMII Mode. Indicates if the FEC is in RMII or MII/7-wire mode. See the chip configuration chapter for more details.<br><br><b>NOTE:</b> Interface selection is chip-specific; see the chapter that describes how modules are configured and connected.<br><br>0 FEC is in true MII mode.<br>1 FEC is in RMII mode.                                                                                                                                                                              |
| 24–25<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                         |
| 26<br>FCE         | Flow control enable. If asserted, the receiver detects PAUSE frames. Upon PAUSE frame detection, the transmitter stops transmitting data frames for a given duration.                                                                                                                                                                                                                                                                                                                           |
| 27<br>BC_REJ      | Broadcast frame reject. If asserted, frames with DA (destination address) equal to FFFF_FFFF_FFFF are rejected unless the PROM bit is set. If BC_REJ and PROM are set, frames with broadcast DA are accepted and the M (MISS) is set in the receive buffer descriptor.                                                                                                                                                                                                                          |
| 28<br>PROM        | Promiscuous mode. All frames are accepted regardless of address matching.                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 29<br>MII_MODE    | Media independent interface mode. Selects the external interface mode for transmit and receive blocks.<br><br><b>NOTE:</b> Interface selection is chip-specific; see the chapter that describes how modules are configured and connected.<br><br>0 7-wire mode (used only for serial 10 Mbps)<br>1 MII or RMII mode as indicated by the RMII_MODE bit                                                                                                                                           |

Table continues on the next page...

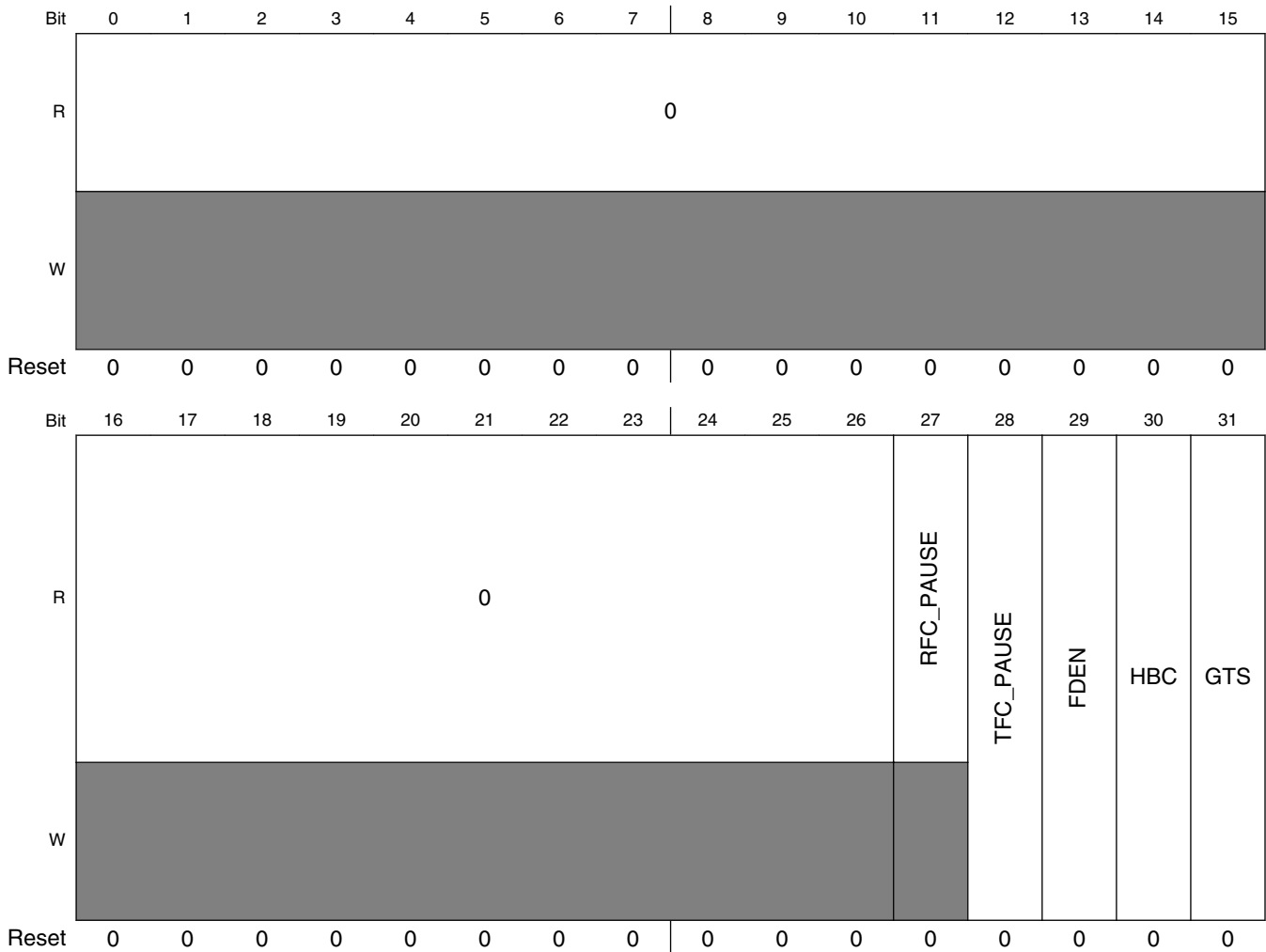
FEC\_RCR field descriptions (continued)

| Field      | Description                                                                                                                                                                                                                                                |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 30<br>DRT  | Disable receive on transmit.<br><br>0 Receive path operates independently of transmit (use for full duplex or to monitor transmit activity in half duplex mode).<br>1 Disable reception of frames while transmitting (normally used for half duplex mode). |
| 31<br>LOOP | Internal loopback. If set, transmitted frames are looped back internal to the device and transmit output signals are not asserted. The internal bus clock substitutes for the FEC_TXCLK when LOOP is asserted. DRT must be set to 0 when setting LOOP.     |

40.5.10 Transmit Control Register (FEC\_TCR)

The TCR configures the transmit block.

Address: 0h base + C4h offset = C4h



## FEC\_TCR field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–26<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 27<br>RFC_PAUSE  | Receive frame control pause. This read-only status bit is asserted when a full duplex flow control pause frame is received and the transmitter pauses for the duration defined in this pause frame. This bit automatically clears when the pause duration is complete.                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 28<br>TFC_PAUSE  | Transmit frame control pause. Transmits a PAUSE frame when asserted. When this bit is set, the MAC stops transmission of data frames after the current transmission is complete. At this time, GRA interrupt in the EIR is asserted. With transmission of data frames stopped, MAC transmits a MAC Control PAUSE frame. Next, the MAC clears the TFC_PAUSE bit and resumes transmitting data frames. If the transmitter pauses due to user assertion of GTS or reception of a PAUSE frame, the MAC may continue transmitting a MAC Control PAUSE frame.                                                                                                                                                    |
| 29<br>FDEN       | Full duplex enable. If set, frames transmit independent of carrier sense and collision inputs. This field must be modified only when ECR[ETHER_EN] = 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 30<br>HBC        | Heartbeat control. If set, the heartbeat check performs following end of transmission and the HB bit in the status register is set if the collision input does not assert within the heartbeat window. This field must be modified only when ECR[ETHER_EN] = 0.                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 31<br>GTS        | Graceful transmit stop. When this bit is set, MAC stops transmission after any frame currently transmitted is complete and GRA interrupt in the EIR is asserted. If frame transmission is not currently underway, the GRA interrupt is asserted immediately. After transmission finishes, clear GTS to restart. The next frame in the transmit FIFO is then transmitted. If an early collision occurs during transmission when GTS is set, transmission stops after the collision. The frame is transmitted again after GTS is cleared. There may be old frames in the transmit FIFO that transmit when GTS is reasserted. To avoid this situation, program ECR[ETHER_EN] = 0 following the GRA interrupt. |

## 40.5.11 Physical Address Low Register (FEC\_PALR)

PALR contains the lower 32 bits (bytes 0, 1, 2, 3) of the 48-bit address used in the address recognition process to compare with the DA (destination address) field of receive frames with an individual DA. In addition, this register is used in bytes 0 through 3 of the 6-byte source address field when transmitting PAUSE frames.

This register is not reset and you must initialize it.

Address: 0h base + E4h offset = E4h

|       |        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|--------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0      | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     | PADDR1 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | u*     | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* |    |

\* Notes:

- u = Unaffected by reset.

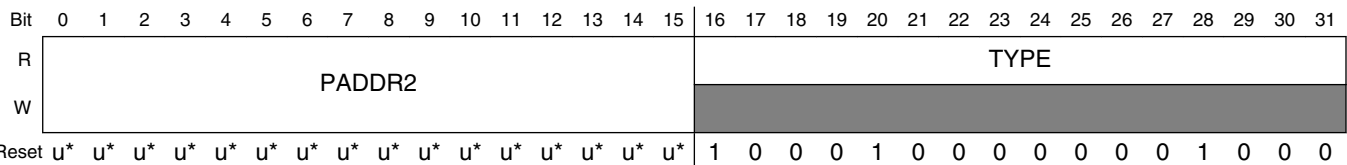
## FEC\_PALR field descriptions

| Field          | Description                                                                                                                                                                   |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>PADDR1 | Bytes 0 (bits 31:24), 1 (bits 23:16), 2 (bits 15:8), and 3 (bits 7:0) of the 6-byte individual address are used for exact match and the source address field in PAUSE frames. |

40.5.12 Physical Address High Register and Type Field (FEC\_PAUR)

PAUR contains the upper 16 bits (bytes 4 and 5) of the 48-bit address used in the address recognition process to compare with the DA (destination address) field of receive frames with an individual DA. In addition, this register is used in bytes 4 and 5 of the 6-byte Source Address field when transmitting PAUSE frames. Bits 15:0 of PAUR contain a constant type field (8808h) for transmission of PAUSE frames. The upper 16 bits of this register are not reset and you must initialize them.

Address: 0h base + E8h offset = E8h



- \* Notes:
- u = Unaffected by reset.

FEC\_PAUR field descriptions

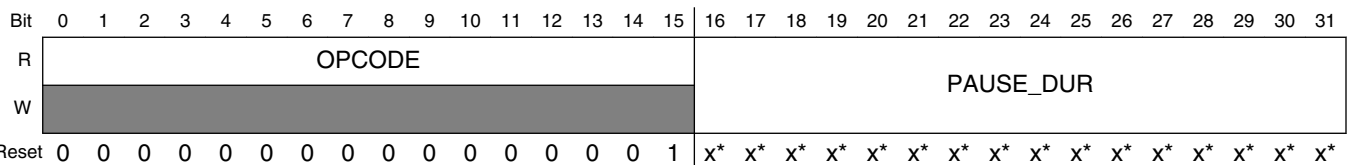
| Field          | Description                                                                                                                                  |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>PADDR2 | Bytes 4 (bits 31:24) and 5 (bits 23:16) of the 6-byte individual address used for exact match, and the source address field in PAUSE frames. |
| 16–31<br>TYPE  | Type field in PAUSE frames. These 16 read-only bits are a constant value of 8808h.                                                           |

40.5.13 Opcode/Pause Duration (FEC\_OPD)

The OPD contains the 16-bit opcode and 16-bit pause duration fields used in transmission of a PAUSE frame. The opcode field is a constant value, 0001h. When another node detects a PAUSE frame, that node pauses transmission for the duration specified in the pause duration field.

The lower 16 bits of this register are not reset and you must initialize them.

Address: 0h base + ECh offset = ECh





\* Notes:

- x = Undefined at reset.

### FEC\_OPD field descriptions

| Field              | Description                                                                    |
|--------------------|--------------------------------------------------------------------------------|
| 0–15<br>OPCODE     | Opcode field used in PAUSE frames. These read-only bits are a constant, 0001h. |
| 16–31<br>PAUSE_DUR | Pause Duration field used in PAUSE frames.                                     |

## 40.5.14 Descriptor Individual Upper Address Register (FEC\_IAUR)

IAUR contains the upper 32 bits of the 64-bit individual address hash table. The address recognition process uses this table to check for a possible match with the destination address (DA) field of receive frames with an individual DA.

This register is not reset and you must initialize it.

Address: 0h base + 118h offset = 118h

|       |        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|--------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0      | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     | IADDR1 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | u*     | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* |    |

\* Notes:

- u = Unaffected by reset.

### FEC\_IAUR field descriptions

| Field          | Description                                                                                                                                                                                                            |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>IADDR1 | The upper 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a unicast address. Bit 31 of IADDR1 contains hash index bit 63. Bit 0 of IADDR1 contains hash index bit 32. |

## 40.5.15 Descriptor Individual Lower Address Register (FEC\_IALR)

IALR contains the lower 32 bits of the 64-bit individual address hash table. The address recognition process uses this table to check for a possible match with the DA field of receive frames with an individual DA.

This register is not reset and you must initialize it.

## Memory map and register definition

Address: 0h base + 11Ch offset = 11Ch

| Bit   | 0      | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|-------|--------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| R     | IADDR2 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | u*     | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* |

\* Notes:

- u = Unaffected by reset.

### FEC\_IALR field descriptions

| Field          | Description                                                                                                                                                                                                           |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>IADDR2 | The lower 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a unicast address. Bit 31 of IADDR2 contains hash index bit 31. Bit 0 of IADDR2 contains hash index bit 0. |

## 40.5.16 Descriptor Group Upper Address Register (FEC\_GAUR)

GAUR contains the upper 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a multicast address.

This register is not reset and you must initialize it.

Address: 0h base + 120h offset = 120h

| Bit   | 0      | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|-------|--------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| R     | GADDR1 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | u*     | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* |

\* Notes:

- u = Unaffected by reset.

### FEC\_GAUR field descriptions

| Field          | Description                                                                                                                                                                                                                                           |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>GADDR1 | The GADDR1 register contains the upper 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a multicast address. Bit 31 of GADDR1 contains hash index bit 63. Bit 0 of GADDR1 contains hash index bit 32. |

## 40.5.17 Descriptor Group Lower Address Register (FEC\_GALR)

GALR contains the lower 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a multicast address.

This register is not reset and you must initialize it.

Address: 0h base + 124h offset = 124h

|       |        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|--------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0      | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | GADDR2 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | u*     | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* |

\* Notes:

- u = Unaffected by reset.

### FEC\_GALR field descriptions

| Field          | Description                                                                                                                                                                                                                                          |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>GADDR2 | The GADDR2 register contains the lower 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a multicast address. Bit 31 of GADDR2 contains hash index bit 31. Bit 0 of GADDR2 contains hash index bit 0. |

## 40.5.18 Transmit FIFO Watermark (FEC\_TFWR)

The TFWR controls the amount of data required in the transmit FIFO before transmission of a frame can begin. This allows you to minimize transmit latency (TFWR = 00 or 01) or allow for larger bus access latency (TFWR = 11) due to contention for the system bus. Setting the watermark to a high value minimizes the risk of transmit FIFO underrun due to contention for the system bus. The byte counts associated with the TFWR field may need to be modified to match a given system requirement (worst case bus access latency by the transmit data DMA channel).

Address: 0h base + 144h offset = 144h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31   |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    | TFWR |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    |

### FEC\_TFWR field descriptions

| Field            | Description                                                                                           |
|------------------|-------------------------------------------------------------------------------------------------------|
| 0–29<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.               |
| 30–31<br>TFWR    | Number of bytes written to transmit FIFO before transmission of a frame begins<br>00 64 bytes written |

Table continues on the next page...

**FEC\_TFWR field descriptions (continued)**

| Field | Description       |
|-------|-------------------|
| 01    | 64 bytes written  |
| 10    | 128 bytes written |
| 11    | 192 bytes written |

**40.5.19 FIFO Receive Bound Register (FEC\_FRBR)**

FRBR indicates the upper address bound of the FIFO RAM. Drivers can use this value, along with the FRSR, to appropriately divide the available FIFO RAM between the transmit and receive data paths.

Address: 0h base + 14Ch offset = 14Ch

|       |    |    |    |    |    |    |         |    |    |    |    |    |    |    |    |    |   |  |
|-------|----|----|----|----|----|----|---------|----|----|----|----|----|----|----|----|----|---|--|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6       | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |   |  |
| R     | 0  |    |    |    |    |    |         |    |    |    |    |    |    |    |    |    |   |  |
| W     |    |    |    |    |    |    |         |    |    |    |    |    |    |    |    |    |   |  |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0       | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |   |  |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22      | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |   |  |
| R     | 0  |    |    |    |    | 1  | R_BOUND |    |    |    |    |    |    |    |    |    | 0 |  |
| W     |    |    |    |    |    |    |         |    |    |    |    |    |    |    |    |    |   |  |
| Reset | 0  | 0  | 0  | 0  | 0  | 1  | 1       | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |   |  |

**FEC\_FRBR field descriptions**

| Field             | Description                                                                             |
|-------------------|-----------------------------------------------------------------------------------------|
| 0–20<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 21<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 1. |
| 22–29<br>R_BOUND  | Highest valid FIFO RAM address.                                                         |
| 30–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |

**40.5.20 FIFO Receive Start Register (FEC\_FRSR)**

FRSR indicates the starting address of the receive FIFO. FRSR marks the boundary between the transmit and receive FIFOs. The transmit FIFO uses addresses from the start of the FIFO to the location four bytes before the address programmed into the FRSR. The receive FIFO uses addresses from FRSR to FRBR inclusive.

Hardware initializes the FRSR at reset. FRSR only needs to be written to change the default value.

Address: 0h base + 150h offset = 150h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |    |    |    |    |    |          |    |    |    |    |    |    |    |    |    |   |  |
|-------|----|----|----|----|----|----|----------|----|----|----|----|----|----|----|----|----|---|--|
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22       | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |   |  |
| R     | 0  |    |    |    |    | 1  | R_FSTART |    |    |    |    |    |    |    |    |    | 0 |  |
| W     |    |    |    |    |    |    |          |    |    |    |    |    |    |    |    |    |   |  |
| Reset | 0  | 0  | 0  | 0  | 0  | 1  | 0        | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |   |  |

### FEC\_FRSR field descriptions

| Field             | Description                                                                                                                                                           |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–20<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                               |
| 21<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 1.                                                                               |
| 22–29<br>R_FSTART | Address of first receive FIFO location. Acts as delimiter between receive and transmit FIFOs.<br>For proper operation, ensure that R_FSTART is set to 48h or greater. |
| 30–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                               |

## 40.5.21 Receive Descriptor Ring Start Register (FEC\_ERDSR)

ERDSR points to the start of the circular receive buffer descriptor queue in external memory. This pointer must be 32-bit aligned; however, it is recommended it be made 128-bit aligned (evenly divisible by 16). You should write zeros to bits 1 and 0. Hardware ignores non-zero values in these two bit positions.

This register is undefined at reset and must be initialized prior to operation.

Address: 0h base + 180h offset = 180h

|       |             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0           | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | R_DES_START |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0  |    |
| W     |             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | u*          | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* |

\* Notes:

- u = Unaffected by reset.

**FEC\_ERDSR field descriptions**

| Field               | Description                                                                             |
|---------------------|-----------------------------------------------------------------------------------------|
| 0–29<br>R_DES_START | Pointer to start of receive buffer descriptor queue.                                    |
| 30–31<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0. |

**40.5.22 Transmit Buffer Descriptor Ring Start Register (FEC\_ETDSR)**

ETSDR provides a pointer to the start of the circular transmit buffer descriptor queue in external memory. This pointer must be 32-bit aligned; however, it is recommended it be made 128-bit aligned (evenly divisible by 16). You should write zeros to bits 1 and 0. Hardware ignores non-zero values in these two bit positions.

This register is undefined at reset and must be initialized prior to operation.

Address: 0h base + 184h offset = 184h

|       |             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0           | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | X_DES_START |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0  |    |
| W     |             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | u*          | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* |    |

\* Notes:

- u = Unaffected by reset.

**FEC\_ETDSR field descriptions**

| Field               | Description                                                                             |
|---------------------|-----------------------------------------------------------------------------------------|
| 0–29<br>X_DES_START | Pointer to start of transmit buffer descriptor queue.                                   |
| 30–31<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0. |

**40.5.23 Receive Buffer Size Register (FEC\_EMRBR)**

The EMRBR dictates the maximum size of all receive buffers. This value should take into consideration that the receive CRC is always written into the last receive buffer. To allow one maximum size frame per buffer, EMRBR must be set to RCR[MAX\_FL] or larger. To properly align the buffer, EMRBR must be evenly divisible by 16. To ensure this, bits 3-0 are forced low.

To minimize bus utilization (descriptor fetches), it is recommended that EMRBR be greater than or equal to 256 bytes.

The EMRBR is undefined at reset and must be initialized by the user.

Address: 0h base + 188h offset = 188h

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17         | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |  |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | R_BUF_SIZE |    |    |    |    |    |    |    |    |    | 0  |    |    |    |    |  |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |
| Reset | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u*         | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* | u* |  |

\* Notes:

- u = Unaffected by reset.

### FEC\_EMRBR field descriptions

| Field               | Description                                                                                                                                                                                                                                                                                                                                                                                                                      |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–20<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                          |
| 21–27<br>R_BUF_SIZE | Maximum size of receive buffer size in bytes. To minimize bus utilization (descriptor fetches), set this field to 256 bytes (10h) or larger.<br><br>10h: 256 + 15 bytes (minimum size recommended)<br><br>11h: 272 + 15 bytes<br><br>...<br><br>7Fh: 2032 + 15 bytes. The FEC writes up to 2047 bytes in the receive buffer. If data larger than 2047 is received, the FEC truncates it and shows 7FFh in the receive descriptor |
| 28–31<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                          |

## 40.5.24 Count of frames not counted correctly (FEC\_RMON\_T\_DROP)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 200h offset = 200h

|       |             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0           | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     | RMON_T_DROP |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*          | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |

\* Notes:

- x = Undefined at reset.

### FEC\_RMON\_T\_DROP field descriptions

| Field               | Description                           |
|---------------------|---------------------------------------|
| 0–31<br>RMON_T_DROP | Count of frames not counted correctly |

## 40.5.25 RMON Tx packet count (FEC\_RMON\_T\_PACKETS)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 204h offset = 204h

|       |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0              | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | RMON_T_PACKETS |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*             | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

### FEC\_RMON\_T\_PACKETS field descriptions

| Field                  | Description          |
|------------------------|----------------------|
| 0–31<br>RMON_T_PACKETS | RMON Tx packet count |

## 40.5.26 RMON Tx broadcast packets (FEC\_RMON\_T\_BC\_PKT)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 208h offset = 208h

|       |               |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0             | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | RMON_T_BC_PKT |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |               |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*            | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

### FEC\_RMON\_T\_BC\_PKT field descriptions

| Field                 | Description               |
|-----------------------|---------------------------|
| 0–31<br>RMON_T_BC_PKT | RMON Tx broadcast packets |



## 40.5.27 RMON Tx multicast packets (FEC\_RMON\_T\_MC\_PKT)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 20Ch offset = 20Ch

|       |               |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0             | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | RMON_T_MC_PKT |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |               |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*            | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

### FEC\_RMON\_T\_MC\_PKT field descriptions

| Field                 | Description               |
|-----------------------|---------------------------|
| 0–31<br>RMON_T_MC_PKT | RMON Tx multicast packets |

## 40.5.28 RMON Tx packets with CRC/align error (FEC\_RMON\_T\_CRC\_ALIGN)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 210h offset = 210h

|       |                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0                | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | RMON_T_CRC_ALIGN |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*               | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

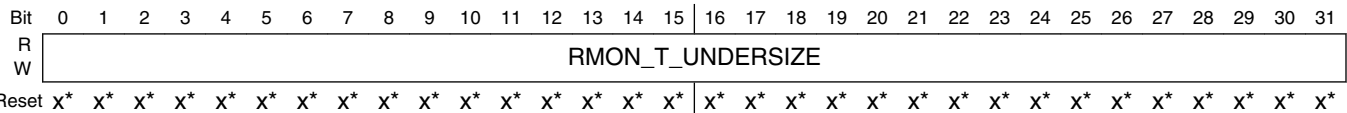
### FEC\_RMON\_T\_CRC\_ALIGN field descriptions

| Field                    | Description                          |
|--------------------------|--------------------------------------|
| 0–31<br>RMON_T_CRC_ALIGN | RMON Tx packets with CRC/align error |

40.5.29 RMON Tx packets < 64 bytes, good CRC (FEC\_RMONT\_UNDERSIZE)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 214h offset = 214h



- \* Notes:
- x = Undefined at reset.

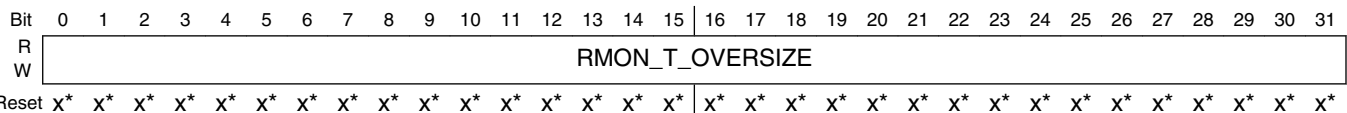
FEC\_RMONT\_UNDERSIZE field descriptions

| Field                   | Description                          |
|-------------------------|--------------------------------------|
| 0–31<br>RMONT_UNDERSIZE | RMON Tx packets < 64 bytes, good CRC |

40.5.30 RMON Tx packets > MAX\_FL bytes, good CRC (FEC\_RMONT\_OVERSIZE)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 218h offset = 218h



- \* Notes:
- x = Undefined at reset.

FEC\_RMONT\_OVERSIZE field descriptions

| Field                  | Description                              |
|------------------------|------------------------------------------|
| 0–31<br>RMONT_OVERSIZE | RMON Tx packets > MAX_FL bytes, good CRC |

### 40.5.31 RMON Tx packets < 64 bytes, bad CRC (FEC\_RMONT\_FRAG)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 21Ch offset = 21Ch

|       |             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0           | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | RMON_T_FRAG |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*          | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

#### FEC\_RMONT\_FRAG field descriptions

| Field              | Description                         |
|--------------------|-------------------------------------|
| 0–31<br>RMONT_FRAG | RMON Tx packets < 64 bytes, bad CRC |

### 40.5.32 RMON Tx packets > MAX\_FL bytes, bad CRC (FEC\_RMONT\_JAB)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 220h offset = 220h

|       |            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0          | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | RMON_T_JAB |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*         | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

#### FEC\_RMONT\_JAB field descriptions

| Field             | Description                             |
|-------------------|-----------------------------------------|
| 0–31<br>RMONT_JAB | RMON Tx packets > MAX_FL bytes, bad CRC |

### 40.5.33 RMON Tx collision count (FEC\_RMON\_T\_COL)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 224h offset = 224h

|       |            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0          | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | RMON_T_COL |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*         | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |

\* Notes:

- x = Undefined at reset.

#### FEC\_RMON\_T\_COL field descriptions

| Field              | Description             |
|--------------------|-------------------------|
| 0–31<br>RMON_T_COL | RMON Tx collision count |

### 40.5.34 RMON Tx 64 byte packets (FEC\_RMON\_T\_P64)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 228h offset = 228h

|       |            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0          | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | RMON_T_P64 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*         | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |

\* Notes:

- x = Undefined at reset.

#### FEC\_RMON\_T\_P64 field descriptions

| Field              | Description             |
|--------------------|-------------------------|
| 0–31<br>RMON_T_P64 | RMON Tx 64 byte packets |

### 40.5.35 RMON Tx 65 to 127 byte packets (FEC\_RMONT\_P65TO127)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 22Ch offset = 22Ch

|       |                 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-----------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0               | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | RMON_T_P65TO127 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |                 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*              | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

#### FEC\_RMONT\_P65TO127 field descriptions

| Field                  | Description                    |
|------------------------|--------------------------------|
| 0–31<br>RMONT_P65TO127 | RMON Tx 65 to 127 byte packets |

### 40.5.36 RMON Tx 128 to 255 byte packets (FEC\_RMONT\_P128TO255)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 230h offset = 230h

|       |                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0                | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | RMON_T_P128TO255 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | X*               | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* |    |

\* Notes:

- x = Undefined at reset.

#### FEC\_RMONT\_P128TO255 field descriptions

| Field                   | Description                     |
|-------------------------|---------------------------------|
| 0–31<br>RMONT_P128TO255 | RMON Tx 128 to 255 byte packets |

### 40.5.37 RMON Tx 256 to 511 byte packets (FEC\_RMONT\_P256TO511)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 234h offset = 234h

|       |                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0                | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | RMON_T_P256TO511 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | X*               | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* |    |

\* Notes:

- x = Undefined at reset.

#### FEC\_RMONT\_P256TO511 field descriptions

| Field                       | Description                     |
|-----------------------------|---------------------------------|
| 0–31<br>RMONT_<br>P256TO511 | RMON Tx 256 to 511 byte packets |

### 40.5.38 RMON Tx 512 to 1023 byte packets (FEC\_RMONT\_P512TO1023)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 238h offset = 238h

|       |                   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0                 | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | RMON_T_P512TO1023 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |                   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*                | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

#### FEC\_RMONT\_P512TO1023 field descriptions

| Field                        | Description                      |
|------------------------------|----------------------------------|
| 0–31<br>RMONT_<br>P512TO1023 | RMON Tx 512 to 1023 byte packets |

### 40.5.39 RMON Tx 1024 to 2047 byte packets (FEC\_RMONT\_P1024TO2047)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 23Ch offset = 23Ch

|       |                    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|--------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0                  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | RMON_T_P1024TO2047 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |                    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | X*                 | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* |    |

\* Notes:

- x = Undefined at reset.

#### FEC\_RMONT\_P1024TO2047 field descriptions

| Field                     | Description                       |
|---------------------------|-----------------------------------|
| 0–31<br>RMONT_P1024TO2047 | RMON Tx 1024 to 2047 byte packets |

### 40.5.40 RMON Tx packets with > 2048 bytes (FEC\_RMONT\_P\_GTE2048)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 240h offset = 240h

|       |                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0                | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | RMON_T_P_GTE2048 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*               | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

#### FEC\_RMONT\_P\_GTE2048 field descriptions

| Field                   | Description                       |
|-------------------------|-----------------------------------|
| 0–31<br>RMONT_P_GTE2048 | RMON Tx packets with > 2048 bytes |

## 40.5.41 RMON Tx Octets (FEC\_RMON\_T\_OCTETS)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 244h offset = 244h

|       |               |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0             | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | RMON_T_OCTETS |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |               |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*            | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

### FEC\_RMON\_T\_OCTETS field descriptions

| Field                 | Description    |
|-----------------------|----------------|
| 0–31<br>RMON_T_OCTETS | RMON Tx Octets |

## 40.5.42 Count of transmitted frames not counted correctly (FEC\_IEEE\_T\_DROP)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 248h offset = 248h

|       |             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0           | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | IEEE_T_DROP |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*          | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

### FEC\_IEEE\_T\_DROP field descriptions

| Field               | Description                                       |
|---------------------|---------------------------------------------------|
| 0–31<br>IEEE_T_DROP | Count of transmitted frames not counted correctly |



### 40.5.43 Frames transmitted OK (FEC\_IEEE\_T\_FRAME\_OK)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 24Ch offset = 24Ch

|       |                 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-----------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0               | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | IEEE_T_FRAME_OK |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |                 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*              | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

#### FEC\_IEEE\_T\_FRAME\_OK field descriptions

| Field                   | Description           |
|-------------------------|-----------------------|
| 0–31<br>IEEE_T_FRAME_OK | Frames transmitted OK |

### 40.5.44 Frames transmitted with single collision (FEC\_IEEE\_T\_1COL)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 250h offset = 250h

|       |             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0           | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | IEEE_T_1COL |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*          | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

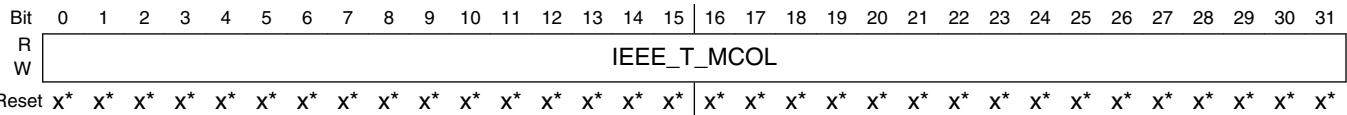
#### FEC\_IEEE\_T\_1COL field descriptions

| Field               | Description                              |
|---------------------|------------------------------------------|
| 0–31<br>IEEE_T_1COL | Frames transmitted with single collision |

40.5.45 Frames transmitted with multiple collisions (FEC\_IEEE\_T\_MCOL)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 254h offset = 254h



- \* Notes:
- x = Undefined at reset.

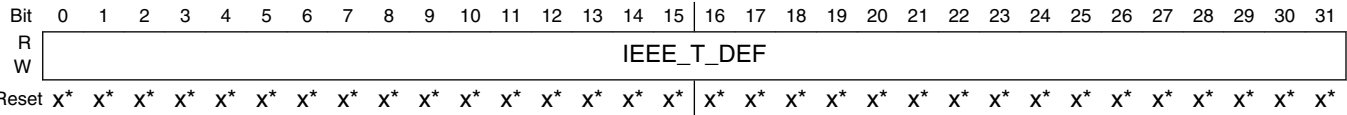
FEC\_IEEE\_T\_MCOL field descriptions

| Field               | Description                                 |
|---------------------|---------------------------------------------|
| 0–31<br>IEEE_T_MCOL | Frames transmitted with multiple collisions |

40.5.46 Frames transmitted after deferral delay (FEC\_IEEE\_T\_DEF)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 258h offset = 258h



- \* Notes:
- x = Undefined at reset.

FEC\_IEEE\_T\_DEF field descriptions

| Field              | Description                             |
|--------------------|-----------------------------------------|
| 0–31<br>IEEE_T_DEF | Frames transmitted after deferral delay |

### 40.5.47 Frames transmitted with late collision (FEC\_IEEE\_T\_LCOL)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 25Ch offset = 25Ch

|       |             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0           | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | IEEE_T_LCOL |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*          | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

#### FEC\_IEEE\_T\_LCOL field descriptions

| Field               | Description                            |
|---------------------|----------------------------------------|
| 0–31<br>IEEE_T_LCOL | Frames transmitted with late collision |

### 40.5.48 Frames transmitted with excessive collisions (FEC\_IEEE\_T\_EXCOL)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 260h offset = 260h

|       |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0            | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | IEEE_T_EXCOL |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*           | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

#### FEC\_IEEE\_T\_EXCOL field descriptions

| Field                | Description                                  |
|----------------------|----------------------------------------------|
| 0–31<br>IEEE_T_EXCOL | Frames transmitted with excessive collisions |

## 40.5.49 Frames transmitted with Tx FIFO underrun (FEC\_IEEE\_T\_MACERR)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 264h offset = 264h

| Bit   | 0             | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|-------|---------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| R     | IEEE_T_MACERR |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |               |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | X*            | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* |    |

\* Notes:

- x = Undefined at reset.

### FEC\_IEEE\_T\_MACERR field descriptions

| Field                 | Description                              |
|-----------------------|------------------------------------------|
| 0–31<br>IEEE_T_MACERR | Frames transmitted with Tx FIFO underrun |

## 40.5.50 Frames transmitted with carrier sense error (FEC\_IEEE\_T\_CSERR)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 268h offset = 268h

| Bit   | 0            | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|-------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| R     | IEEE_T_CSERR |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*           | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

### FEC\_IEEE\_T\_CSERR field descriptions

| Field                | Description                                 |
|----------------------|---------------------------------------------|
| 0–31<br>IEEE_T_CSERR | Frames transmitted with carrier sense error |

### 40.5.51 Frames transmitted with SQE error (FEC\_IEEE\_T\_SQE)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 26Ch offset = 26Ch

|       |            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0          | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | IEEE_T_SQE |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*         | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |

\* Notes:

- x = Undefined at reset.

#### FEC\_IEEE\_T\_SQE field descriptions

| Field              | Description                       |
|--------------------|-----------------------------------|
| 0–31<br>IEEE_T_SQE | Frames transmitted with SQE error |

### 40.5.52 Flow control pause frames transmitted (FEC\_IEEE\_T\_FDXFC)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 270h offset = 270h

|       |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0            | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | IEEE_T_FDXFC |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*           | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |

\* Notes:

- x = Undefined at reset.

#### FEC\_IEEE\_T\_FDXFC field descriptions

| Field                | Description                           |
|----------------------|---------------------------------------|
| 0–31<br>IEEE_T_FDXFC | Flow control pause frames transmitted |

### 40.5.53 Octet count for frames transmitted without error (FEC\_IEEE\_T\_OCTETS\_OK)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 274h offset = 274h

|       |                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0                | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | IEEE_T_OCTETS_OK |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | X*               | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* |    |

\* Notes:

- x = Undefined at reset.

#### FEC\_IEEE\_T\_OCTETS\_OK field descriptions

| Field                    | Description                                      |
|--------------------------|--------------------------------------------------|
| 0–31<br>IEEE_T_OCTETS_OK | Octet count for frames transmitted without error |

### 40.5.54 Count of received frames not counted correctly (FEC\_RMON\_R\_DROP)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 280h offset = 280h

|       |             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0           | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | RMON_R_DROP |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | X*          | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* |    |

\* Notes:

- x = Undefined at reset.

#### FEC\_RMON\_R\_DROP field descriptions

| Field               | Description                                    |
|---------------------|------------------------------------------------|
| 0–31<br>RMON_R_DROP | Count of received frames not counted correctly |

### 40.5.55 RMON Rx packet count (FEC\_RMON\_R\_PACKETS)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 284h offset = 284h

|       |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0              | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | RMON_R_PACKETS |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*             | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

#### FEC\_RMON\_R\_PACKETS field descriptions

| Field                  | Description          |
|------------------------|----------------------|
| 0–31<br>RMON_R_PACKETS | RMON Rx packet count |

### 40.5.56 RMON Rx broadcast packets (FEC\_RMON\_R\_BC\_PKT)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 288h offset = 288h

|       |               |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0             | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | RMON_R_BC_PKT |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |               |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*            | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

#### FEC\_RMON\_R\_BC\_PKT field descriptions

| Field                 | Description               |
|-----------------------|---------------------------|
| 0–31<br>RMON_R_BC_PKT | RMON Rx broadcast packets |

## 40.5.57 RMON Rx multicast packets (FEC\_RMON\_R\_MC\_PKT)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 28Ch offset = 28Ch

|       |               |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0             | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | RMON_R_MC_PKT |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |               |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*            | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

### FEC\_RMON\_R\_MC\_PKT field descriptions

| Field                 | Description               |
|-----------------------|---------------------------|
| 0–31<br>RMON_R_MC_PKT | RMON Rx multicast packets |

## 40.5.58 RMON Rx packets with CRC/Align error (FEC\_RMON\_R\_CRC\_ALIGN)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 290h offset = 290h

|       |                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0                | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | RMON_R_CRC_ALIGN |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*               | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

### FEC\_RMON\_R\_CRC\_ALIGN field descriptions

| Field                    | Description                          |
|--------------------------|--------------------------------------|
| 0–31<br>RMON_R_CRC_ALIGN | RMON Rx packets with CRC/Align error |



### 40.5.59 RMON Rx packets < 64 bytes, good CRC (FEC\_RMONT\_R\_UNDERSIZE)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 294h offset = 294h

|       |                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0                | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | RMON_R_UNDERSIZE |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*               | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

#### FEC\_RMONT\_R\_UNDERSIZE field descriptions

| Field                     | Description                           |
|---------------------------|---------------------------------------|
| 0–31<br>RMONT_R_UNDERSIZE | RMONT Rx packets < 64 bytes, good CRC |

### 40.5.60 RMON Rx packets > MAX\_FL bytes, good CRC (FEC\_RMONT\_R\_OVERSIZE)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 298h offset = 298h

|       |                 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-----------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0               | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | RMON_R_OVERSIZE |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |                 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*              | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

#### FEC\_RMONT\_R\_OVERSIZE field descriptions

| Field                    | Description                               |
|--------------------------|-------------------------------------------|
| 0–31<br>RMONT_R_OVERSIZE | RMONT Rx packets > MAX_FL bytes, good CRC |

## 40.5.61 RMON Rx packets < 64 bytes, bad CRC (FEC\_RMOM\_R\_FRAG)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 29Ch offset = 29Ch

|       |             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0           | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | RMON_R_FRAG |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | X*          | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* |    |

\* Notes:

- x = Undefined at reset.

### FEC\_RMOM\_R\_FRAG field descriptions

| Field               | Description                         |
|---------------------|-------------------------------------|
| 0–31<br>RMOM_R_FRAG | RMON Rx packets < 64 bytes, bad CRC |

## 40.5.62 RMON Rx packets > MAX\_FL bytes, bad CRC (FEC\_RMOM\_R\_JAB)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 2A0h offset = 2A0h

|       |            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0          | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | RMON_R_JAB |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*         | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

### FEC\_RMOM\_R\_JAB field descriptions

| Field              | Description                             |
|--------------------|-----------------------------------------|
| 0–31<br>RMOM_R_JAB | RMON Rx packets > MAX_FL bytes, bad CRC |

### 40.5.63 Reserved (FEC\_RMON\_R\_RESVD\_0)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 2A4h offset = 2A4h

|       |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0              | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | RMON_R_RESVD_0 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*             | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

#### FEC\_RMON\_R\_RESVD\_0 field descriptions

| Field                      | Description |
|----------------------------|-------------|
| 0–31<br>RMON_R_<br>RESVD_0 | Reserved    |

### 40.5.64 RMON Rx 64 byte packets (FEC\_RMON\_R\_P64)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 2A8h offset = 2A8h

|       |            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0          | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | RMON_R_P64 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*         | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

#### FEC\_RMON\_R\_P64 field descriptions

| Field              | Description             |
|--------------------|-------------------------|
| 0–31<br>RMON_R_P64 | RMON Rx 64 byte packets |

## 40.5.65 RMON Rx 65 to 127 byte packets (FEC\_RMON\_R\_P65TO127)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 2ACh offset = 2ACh

|       |                 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-----------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0               | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | RMON_R_P65TO127 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |                 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*              | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

### FEC\_RMON\_R\_P65TO127 field descriptions

| Field                       | Description                    |
|-----------------------------|--------------------------------|
| 0–31<br>RMON_R_<br>P65TO127 | RMON Rx 65 to 127 byte packets |

## 40.5.66 RMON Rx 128 to 255 byte packets (FEC\_RMON\_R\_P128TO255)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 2B0h offset = 2B0h

|       |                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0                | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | RMON_R_P128TO255 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*               | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

### FEC\_RMON\_R\_P128TO255 field descriptions

| Field                        | Description                     |
|------------------------------|---------------------------------|
| 0–31<br>RMON_R_<br>P128TO255 | RMON Rx 128 to 255 byte packets |

### 40.5.67 RMON Rx 256 to 511 byte packets (FEC\_RMONG\_R\_P256TO511)

See [Using the RMONG and IEEE registers](#).

Address: 0h base + 2B4h offset = 2B4h

|       |                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0                | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | RMON_R_P256TO511 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*               | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

#### FEC\_RMONG\_R\_P256TO511 field descriptions

| Field                         | Description                      |
|-------------------------------|----------------------------------|
| 0–31<br>RMONG_R_<br>P256TO511 | RMONG Rx 256 to 511 byte packets |

### 40.5.68 RMONG Rx 512 to 1023 byte packets (FEC\_RMONG\_R\_P512TO1023)

See [Using the RMONG and IEEE registers](#).

Address: 0h base + 2B8h offset = 2B8h

|       |                   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0                 | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | RMON_R_P512TO1023 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |                   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*                | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

#### FEC\_RMONG\_R\_P512TO1023 field descriptions

| Field                          | Description                       |
|--------------------------------|-----------------------------------|
| 0–31<br>RMONG_R_<br>P512TO1023 | RMONG Rx 512 to 1023 byte packets |

## 40.5.69 RMON Rx 1024 to 2047 byte packets (FEC\_RMOM\_R\_P1024TO2047)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 2BCh offset = 2BCh

|       |                    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|--------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0                  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | RMON_R_P1024TO2047 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |                    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | X*                 | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* |    |

\* Notes:

- x = Undefined at reset.

### FEC\_RMOM\_R\_P1024TO2047 field descriptions

| Field                          | Description                       |
|--------------------------------|-----------------------------------|
| 0–31<br>RMOM_R_<br>P1024TO2047 | RMOM Rx 1024 to 2047 byte packets |

## 40.5.70 RMON Rx packets with > 2048 bytes (FEC\_RMOM\_R\_P\_GTE2048)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 2C0h offset = 2C0h

|       |                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0                | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | RMON_R_P_GTE2048 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*               | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

### FEC\_RMOM\_R\_P\_GTE2048 field descriptions

| Field                        | Description                       |
|------------------------------|-----------------------------------|
| 0–31<br>RMOM_R_P_<br>GTE2048 | RMOM Rx packets with > 2048 bytes |

### 40.5.71 RMON Rx octets (FEC\_RMON\_R\_OCTETS)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 2C4h offset = 2C4h

|       |               |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0             | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | RMON_R_OCTETS |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |               |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*            | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

#### FEC\_RMON\_R\_OCTETS field descriptions

| Field                 | Description    |
|-----------------------|----------------|
| 0–31<br>RMON_R_OCTETS | RMON Rx octets |

### 40.5.72 Count of received frames not counted correctly (FEC\_IEEE\_R\_DROP)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 2C8h offset = 2C8h

|       |             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0           | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | IEEE_R_DROP |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | X*          | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* |    |

\* Notes:

- x = Undefined at reset.

#### FEC\_IEEE\_R\_DROP field descriptions

| Field               | Description                                    |
|---------------------|------------------------------------------------|
| 0–31<br>IEEE_R_DROP | Count of received frames not counted correctly |

### 40.5.73 Frames received OK (FEC\_IEEE\_R\_FRAME\_OK)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 2CCh offset = 2CCh

|       |                 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-----------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0               | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | IEEE_R_FRAME_OK |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |                 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*              | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

#### FEC\_IEEE\_R\_FRAME\_OK field descriptions

| Field                   | Description        |
|-------------------------|--------------------|
| 0–31<br>IEEE_R_FRAME_OK | Frames received OK |

### 40.5.74 Frames received with CRC error (FEC\_IEEE\_R\_CRC)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 2D0h offset = 2D0h

|       |            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0          | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | IEEE_R_CRC |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*         | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

#### FEC\_IEEE\_R\_CRC field descriptions

| Field              | Description                    |
|--------------------|--------------------------------|
| 0–31<br>IEEE_R_CRC | Frames received with CRC error |



### 40.5.75 Frames received with alignment error (FEC\_IEEE\_R\_ALIGN)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 2D4h offset = 2D4h

|       |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0            | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | IEEE_R_ALIGN |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*           | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |

\* Notes:

- x = Undefined at reset.

#### FEC\_IEEE\_R\_ALIGN field descriptions

| Field                | Description                          |
|----------------------|--------------------------------------|
| 0–31<br>IEEE_R_ALIGN | Frames received with alignment error |

### 40.5.76 Receive FIFO overflow count (FEC\_IEEE\_R\_MACERR)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 2D8h offset = 2D8h

|       |               |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0             | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | IEEE_R_MACERR |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |               |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*            | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |

\* Notes:

- x = Undefined at reset.

#### FEC\_IEEE\_R\_MACERR field descriptions

| Field                 | Description                 |
|-----------------------|-----------------------------|
| 0–31<br>IEEE_R_MACERR | Receive FIFO overflow count |

## 40.5.77 Flow control pause frames received (FEC\_IEEE\_R\_FDXFC)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 2DCh offset = 2DCh

|       |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0            | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | IEEE_R_FDXFC |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*           | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

### FEC\_IEEE\_R\_FDXFC field descriptions

| Field                | Description                        |
|----------------------|------------------------------------|
| 0–31<br>IEEE_R_FDXFC | Flow control pause frames received |

## 40.5.78 Octet count for frames received without error (FEC\_IEEE\_R\_OCTETS\_OK)

See [Using the RMON and IEEE registers](#).

Address: 0h base + 2E0h offset = 2E0h

|       |                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0                | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | IEEE_R_OCTETS_OK |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*               | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

### FEC\_IEEE\_R\_OCTETS\_OK field descriptions

| Field                    | Description                                   |
|--------------------------|-----------------------------------------------|
| 0–31<br>IEEE_R_OCTETS_OK | Octet count for frames received without error |

## 40.5.79 Using the RMON and IEEE registers

In order to use the RMON and IEEE registers to monitor FEC performance, you must prepare them using the sequence below:

1. Write zeros to the registers.
2. Enable the FEC.

Until you complete this sequence, the registers are read/write, and their reset value is undefined. After you complete this sequence, the registers become read-only, and you may read them as necessary.

## 40.6 Functional description

This section describes the operation of the FEC, beginning with the description of two different types of MII frames, the buffer descriptors, the hardware and software initialization sequence, then the software (Ethernet driver) interface for transmitting and receiving frames.

Following the software initialization and operation sections are sections providing a detailed description of the functions of the FEC.

### 40.6.1 MII data frame

Ethernet/802.3 data frames transmitted across the MII have the format specified in [Figure 40-81](#) and [Table 40-85](#).

<inter-frame> <preamble> <sfd> <data> <efd>

**Figure 40-81. MII data frame format**

**Table 40-85. MII data frame components**

| Component     | Description                                                                                                                                                                           |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <inter-frame> | Inter-frame period. This is an unspecified amount of time during which no data activity occurs on the MII. The de-assertion of RX_DV and TX_EN indicate the absence of data activity. |
| <preamble>    | Preamble that begins a frame. It has the following value:<br>10101010 10101010 10101010 10101010 10101010 10101010 10101010<br>The leftmost 1 represents the LSB of the byte.         |
| <sfd>         | Start of a frame. It has the value 10101011.                                                                                                                                          |
| <data>        | Data portion of the frame. It consists of N octets which corresponds to 2N nibbles being transmitted. The order of each nibble is defined in <a href="#">Figure 40-82</a> .           |
| <efd>         | End-of-frame delimiter.                                                                                                                                                               |

Table 40-85. MII data frame components

| Component | Description                                                                                                                                      |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------|
|           | For TXD data, it is indicated by the de-assertion of the TX_EN signal.<br>For RXD data, it is indicated by the de-assertion of the RX_DV signal. |

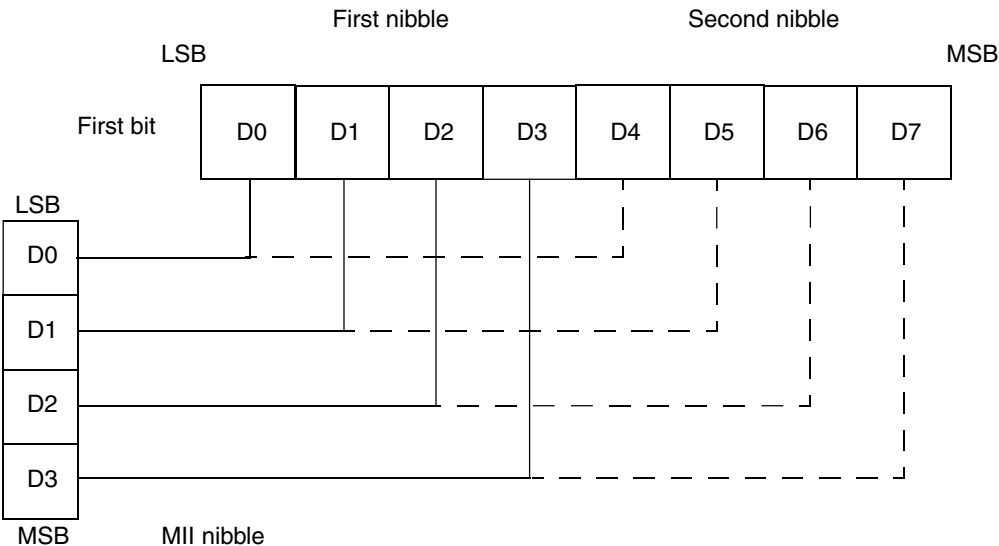


Figure 40-82. MII nibble/octet to octet/nibble mapping

40.6.2 MII management frame structure

A transceiver management frame being transmitted on the MII management interface uses the MDIO and MDC pins. A transaction or frame on this serial interface has the format specified in [Figure 40-83](#) and [Table 40-86](#).

<preamble> <st> <op> <phyad> <regad> <ta> <data> <idle>

Figure 40-83. MII management frame format

Table 40-86. MII management frame components

| Component  | Description                                                                                                            |
|------------|------------------------------------------------------------------------------------------------------------------------|
| <preamble> | Optional preamble, consisting of a sequence of 32 continuous logic 1's.                                                |
| <st>       | Start of frame, consisting of a 01 pattern.                                                                            |
| <op>       | Operation code.<br>01 Write operation<br>10 Read instruction                                                           |
| <phyad>    | Five-bit field that allows up to 32 PHYs to be addressed. The first address bit transmitted is the MSB of the address. |

Table continues on the next page...

**Table 40-86. MII management frame components (continued)**

| Component | Description                                                                                                                                               |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <regad>   | Five-bit field that allows for 32 registers to be addressed within each PHY. The first register bit transmitted is the MSB of the address.                |
| <ta>      | Two-bit field that provides spacing between the register address field and the data field to avoid contention on the MDIO signal during a read operation. |
| <data>    | 16-bit data field. Bit 15 is the first data bit transmitted and received.                                                                                 |
| <idle>    | Idle condition (MDIO is in the high-impedance state).                                                                                                     |

The MII management register set located in the PHY may consist of a basic register set and an extended register set as defined in [Table 40-87](#).

**Table 40-87. MII management register set**

| Register address | Register name                  | Register set |
|------------------|--------------------------------|--------------|
| 0                | Control                        | Basic        |
| 1                | Status                         |              |
| 2-3              | PHY identifier                 | Extended     |
| 4                | Auto-negotiation advertisement |              |
| 5                | AN link partner ability        |              |
| 6                | AN expansion                   |              |
| 7                | AN next page transmit          |              |
| 8-15             | Reserved                       |              |
| 16-31            | Vendor-specific                |              |

### 40.6.3 Buffer descriptors

This section provides a description of the operation of the driver/DMA via the buffer descriptors. It is followed by a detailed description of the receive and transmit descriptor fields.

#### 40.6.3.1 Driver/DMA operation with buffer descriptors

The data for the FEC frames resides in one or more memory buffers external to the FEC. Associated with each buffer is a buffer descriptor (BD), which contains a starting address (32-bit aligned pointer), data length, and status/control information (which contains the current state for the buffer). To permit maximum user flexibility, the BDs are also located in external memory and are read by the FEC DMA engine.

Software produces buffers by allocating/initializing memory and initializing buffer descriptors. Setting the RxBD[E] or TxBD[R] bit produces the buffer. Software writing to TDAR or RDAR tells the FEC that a buffer is placed in external memory for the transmit or receive data traffic, respectively. The hardware reads the BDs and consumes the buffers after they have been produced. After the data DMA is complete and the DMA engine writes the buffer descriptor status bits, hardware clears RxBD[E] or TxBD[R] to signal the buffer has been consumed. Software may poll the BDs to detect when the buffers are consumed or may rely on the buffer/frame interrupts. The driver may process these buffers, and they can return to the free list.

The ECR[ETHER\_EN] bit operates as a reset to the BD/DMA logic. When ECR[ETHER\_EN] is cleared, the DMA engine BD pointers are reset to point to the starting transmit and receive BDs. The buffer descriptors are not initialized by hardware during reset. At least one transmit and receive buffer descriptor must be initialized by software before ECR[ETHER\_EN] is set.

The buffer descriptors operate as two separate rings. ERDSR defines the starting address for receive BDs and ETDSR defines the starting address for transmit BDs. The wrap (W) bit defines the last buffer descriptor in each ring. When W is set, the next descriptor in the ring is at the location pointed to by ERDSR and ETDSR for the receive and transmit rings, respectively. Buffer descriptor rings must start on a 32-bit boundary; however, it is recommended they be 128-bit aligned.

#### 40.6.3.1.1 Driver/DMA operation with transmit BDs

Typically, a transmit frame is divided between multiple buffers. An example is to have an application payload in one buffer, TCP header in a second buffer, IP header in a third buffer, and Ethernet/IEEE 802.3 header in a fourth buffer. The Ethernet MAC does not prepend the Ethernet header (destination address, source address, length/type field(s)), so the driver must provide this in one of the transmit buffers. The Ethernet MAC can append the Ethernet CRC to the frame. TxBD[TC], which must be set by the driver, determines whether the MAC or driver appends the CRC.

The driver (TxBD software producer) should set up Tx BDs so a complete transmit frame is given to the hardware at once. If a transmit frame consists of three buffers, the BDs should be initialized with pointer, length, and control (W, L, TC, ABC) and then the TxBD[R] bit should be set in reverse order (third, second, then first BD) to ensure that the complete frame is ready in memory before the DMA begins. If the TxBDs are set up in order, the DMA controller could DMA the first BD before the second was made available, potentially causing a transmit FIFO underrun.

In the FEC, the driver notifies the DMA that new transmit frame(s) are available by writing to TDAR. When this register is written to (data value is not significant), the FEC RISC tells the DMA to read the next transmit BD in the ring. After started, the RISC + DMA continues to read and interpret transmit BDs in order and DMA the associated buffers until a transmit BD is encountered with the R bit cleared. At this point, the FEC polls this BD one more time. If the R bit is cleared the second time, RISC stops the transmit descriptor read process until software sets up another transmit frame and writes to TDAR.

When the DMA of each transmit buffer is complete, the DMA writes back to the BD to clear the R bit, indicating that the hardware consumer is finished with the buffer.

#### 40.6.3.1.2 Driver/DMA operation with receive BDs

Unlike transmit, the length of the receive frame is unknown by the driver ahead of time. Therefore, the driver must set a variable to define the length of all receive buffers. In the FEC, this variable is written to the EMRBR register.

The driver (RxBD software producer) should set up some number of empty buffers for the Ethernet by initializing the address field and the E and W bits of the associated receive BDs. The hardware (receive DMA) consumes these buffers by filling them with data as frames are received and clearing the E bit and writing to the L bit (1 indicates last buffer in frame), the frame status bits (if L is set), and the length field.

If a receive frame spans multiple receive buffers, the L bit is only set for the last buffer in the frame. For non-last buffers, the length field in the receive BD is written by the DMA (at the same time the E bit is cleared) with the default receive buffer length value. For end-of-frame buffers, the receive BD is written with L set and information written to the status bits (M, BC, MC, LG, NO, CR, OV, TR). Some of the status bits are error indicators which, if set, indicate the receive frame should be discarded and not given to higher layers. The frame status/length information is written into the receive FIFO following the end of the frame (as a single 32-bit word) by the receive logic. The length field for the end of frame buffer is written with the length of the entire frame, not only the length of the last buffer.

For simplicity, the driver may assign a large enough default receive buffer length to contain an entire frame, keeping in mind that a malfunction on the network or out-of-spec implementation could result in giant frames. Frames of 2K (2048) bytes or larger are truncated by the FEC at 2047 bytes so software never sees a receive frame larger than 2047 bytes.

Similar to transmit, the FEC polls the receive descriptor ring after the driver sets up receive BDs and writes to the RDAR register. As frames are received, the FEC fills receive buffers and updates the associated BDs, then reads the next BD in the receive

descriptor ring. If the FEC reads a receive BD and finds the E bit cleared, it polls this BD once more. If RxB<sub>D</sub>[E] is clear a second time, FEC stops reading receive BDs until the driver writes to RDAR.

### 40.6.3.2 Ethernet Receive Buffer Descriptor (RxB<sub>D</sub>)

In the RxB<sub>D</sub>, the user initializes the E and W bits in the first longword and the pointer in the second longword. When the buffer has been DMA'd, the Ethernet controller modifies the E, L, M, BC, MC, LG, NO, CR, OV, and TR bits and writes the length of the used portion of the buffer in the first longword. The M, BC, MC, LG, NO, CR, OV, and TR bits in the first longword of the buffer descriptor are only modified by the Ethernet controller when the L bit is set.

**Table 40-88. Receive Buffer Descriptor (RxB<sub>D</sub>)**

|          | 15                                | 14  | 13 | 12  | 11 | 10 | 9 | 8 | 7  | 6  | 5  | 4  | 3 | 2  | 1  | 0  |
|----------|-----------------------------------|-----|----|-----|----|----|---|---|----|----|----|----|---|----|----|----|
| Offset+0 | E                                 | RO1 | W  | RO2 | L  | —  | — | M | BC | MC | LG | NO | — | CR | OV | TR |
| Offset+2 | Data Length                       |     |    |     |    |    |   |   |    |    |    |    |   |    |    |    |
| Offset+4 | Rx Data Buffer Pointer - A[31:16] |     |    |     |    |    |   |   |    |    |    |    |   |    |    |    |
| Offset+6 | Rx Data Buffer Pointer - A[15:0]  |     |    |     |    |    |   |   |    |    |    |    |   |    |    |    |

**Table 40-89. Receive Buffer Descriptor field descriptions**

| Word       | Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                       |
|------------|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Offset + 0 | E     | Empty. Written by the FEC (=0) and user (=1).<br>0: The data buffer associated with this BD is filled with received data, or data reception has aborted due to an error condition. The status and length fields have been updated as required.<br>1: The data buffer associated with this BD is empty, or reception is currently in progress.                                                                                     |
| Offset + 0 | RO1   | Receive software ownership. This field is reserved for use by software. This read/write bit is not modified by hardware, nor does its value affect hardware.                                                                                                                                                                                                                                                                      |
| Offset + 0 | W     | Wrap. Written by user.<br>0: The next buffer descriptor is found in the consecutive location<br>1: The next buffer descriptor is found at the location defined in ERDSR.                                                                                                                                                                                                                                                          |
| Offset + 0 | RO2   | Receive software ownership. This field is reserved for use by software. This read/write bit is not modified by hardware, nor does its value affect hardware.                                                                                                                                                                                                                                                                      |
| Offset + 0 | L     | Last in frame. Written by the FEC.<br>0: The buffer is not the last in a frame.<br>1: The buffer is the last in a frame.                                                                                                                                                                                                                                                                                                          |
| Offset + 0 | M     | Miss. Written by the FEC. This bit is set by the FEC for frames accepted in promiscuous mode, but flagged as a miss by the internal address recognition. Therefore, while in promiscuous mode, you can use the M-bit to quickly determine whether the frame was destined to this station. This bit is valid only if the L-bit is set and the PROM bit is set.<br>0: The frame was received because of an address recognition hit. |

*Table continues on the next page...*



**Table 40-89. Receive Buffer Descriptor field descriptions (continued)**

| Word       | Field       | Description                                                                                                                                                                                                                                                                                             |
|------------|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|            |             | 1: The frame was received because of promiscuous mode.                                                                                                                                                                                                                                                  |
| Offset + 0 | BC          | Set if the DA is broadcast (FFFF_FFFF_FFFF).                                                                                                                                                                                                                                                            |
| Offset + 0 | MC          | Set if the DA is multicast and not BC.                                                                                                                                                                                                                                                                  |
| Offset + 0 | LG          | Rx frame length violation. Written by the FEC. A frame length greater than RCR[MAX_FL] was recognized. This bit is valid only if the L-bit is set. The receive data is not altered in any way unless the length exceeds 2047 bytes.                                                                     |
| Offset + 0 | NO          | Receive non-octet aligned frame. Written by the FEC. A frame that contained a number of bits not divisible by 8 was received, and the CRC check that occurred at the preceding byte boundary generated an error. This bit is valid only if the L-bit is set. If this bit is set, the CR bit is not set. |
| Offset + 0 | CR          | Receive CRC error. Written by the FEC. This frame contains a CRC error and is an integral number of octets in length. This bit is valid only if the L-bit is set.                                                                                                                                       |
| Offset + 0 | OV          | Overrun. Written by the FEC. A receive FIFO overrun occurred during frame reception. If this bit is set, the other status bits, M, LG, NO, CR, and CL lose their normal meaning and are zero. This bit is valid only if the L-bit is set.                                                               |
| Offset + 0 | TR          | Set if the receive frame is truncated (frame length > 2047 bytes). If the TR bit is set, the frame must be discarded and the other error bits must be ignored as they may be incorrect.                                                                                                                 |
| Offset + 2 | Data Length | Data length. Written by the FEC. Data length is the number of octets written by the FEC into this BD's data buffer if L equals 0 (the value is equal to EMRBR), or the length of the frame including CRC if L is set. It is written by the FEC once as the BD is closed.                                |
| Offset + 4 | A[31:16]    | RX data buffer pointer, bits [31:16] <sup>1</sup>                                                                                                                                                                                                                                                       |
| Offset + 6 | A[15:0]     | RX data buffer pointer, bits [15:0]                                                                                                                                                                                                                                                                     |

1. The receive buffer pointer, containing the address of the associated data buffer, must always be evenly divisible by 16. The buffer must reside in memory external to the FEC. The Ethernet controller never modifies this value.

### Note

When the software driver sets an E bit in one or more receive descriptors, the driver should follow with a write to RDAR.

### 40.6.3.3 Ethernet Transmit Buffer Descriptor (TxBD)

Data is presented to the FEC for transmission by arranging it in buffers referenced by the channel's TxBDs. The Ethernet controller confirms transmission by clearing the ready bit (TxBD[R]) when DMA of the buffer is complete. In the TxBD, the user initializes the R, W, L, and TC bits and the length (in bytes) in the first longword and the buffer pointer in the second longword.

The FEC clears the R bit when the buffer is transferred. Status bits for the buffer/frame are not included in the transmit buffer descriptors. Transmit frame status is indicated via individual interrupt bits (error conditions) and in statistic counters in the MIB block.

**Table 40-90. Transmit Buffer Descriptor (TxBD)**

|          | 15                                | 14  | 13 | 12  | 11 | 10 | 9   | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|----------|-----------------------------------|-----|----|-----|----|----|-----|---|---|---|---|---|---|---|---|---|
| Offset+0 | R                                 | TO1 | W  | TO2 | L  | TC | ABC | — | — | — | — | — | — | — | — | — |
| Offset+2 | Data Length                       |     |    |     |    |    |     |   |   |   |   |   |   |   |   |   |
| Offset+4 | Tx Data Buffer Pointer - A[31:16] |     |    |     |    |    |     |   |   |   |   |   |   |   |   |   |
| Offset+6 | Tx Data Buffer Pointer - A[15:0]  |     |    |     |    |    |     |   |   |   |   |   |   |   |   |   |

**Table 40-91. Transmit Buffer Descriptor field descriptions**

| Word       | Field       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|------------|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Offset + 0 | R           | Ready. Written by the FEC and you.<br><br>0: The data buffer associated with this BD is not ready for transmission. You are free to manipulate this BD or its associated data buffer. The FEC clears this bit after the buffer has been transmitted or after an error condition is encountered.<br><br>1: The data buffer, prepared for transmission by you, has not been transmitted or currently transmits. You may write no fields of this BD after this bit is set. |
| Offset + 0 | TO1         | Transmit software ownership. This field is reserved for software use. This read/write bit is not modified by hardware nor does its value affect hardware.                                                                                                                                                                                                                                                                                                               |
| Offset + 0 | W           | Wrap. Written by user.<br><br>0: The next buffer descriptor is found in the consecutive location<br>1: The next buffer descriptor is found at the location defined in ETDSR.                                                                                                                                                                                                                                                                                            |
| Offset + 0 | TO2         | Transmit software ownership. This field is reserved for use by software. This read/write bit is not modified by hardware nor does its value affect hardware.                                                                                                                                                                                                                                                                                                            |
| Offset + 0 | L           | Last in frame. Written by user.<br><br>0: The buffer is not the last in the transmit frame<br>1: The buffer is the last in the transmit frame                                                                                                                                                                                                                                                                                                                           |
| Offset + 0 | TC          | Transmit CRC. Written by user (only valid if L is set).<br><br>0: End transmission immediately after the last data byte<br>1: Transmit the CRC sequence after the last data byte                                                                                                                                                                                                                                                                                        |
| Offset + 0 | ABC         | Append bad CRC. Written by user (only valid if L is set).<br><br>0: No effect<br>1: Transmit the CRC sequence inverted after the last data byte (regardless of TC value)                                                                                                                                                                                                                                                                                                |
| Offset + 2 | Data Length | Data length, written by user.<br><br>Data length is the number of octets the FEC should transmit from this BD's data buffer. It is never modified by the FEC.                                                                                                                                                                                                                                                                                                           |
| Offset + 4 | A[31:16]    | Tx data buffer pointer, bits [31:16]<br><br>1                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Offset + 6 | A[15:0]     | Tx data buffer pointer, bits [15:0]                                                                                                                                                                                                                                                                                                                                                                                                                                     |

1. The transmit buffer pointer, containing the address of the associated data buffer, must always be evenly divisible by 4. The buffer must reside in memory external to the FEC. This value is never modified by the Ethernet controller.

### Note

After the software driver has set up the buffers for a frame, it should set up the corresponding BDs. The last step in setting up the BDs for a transmit frame is setting the R bit in the first BD for the frame. The driver must follow that with a write to TDAR that triggers the FEC to poll the next BD in the ring.

## 40.6.4 Initialization sequence

This section describes which registers are reset due to hardware reset, which registers are reset by the FEC RISC, and what locations you must initialize prior to enabling the FEC.

### 40.6.4.1 Hardware Controlled Initialization

In the FEC, hardware resets registers and control logic that generate interrupts. A hardware reset negates output signals and resets general configuration bits.

Other registers reset when the ECR[ETHER\_EN] bit is cleared (which is accomplished by a hard reset or software to halt operation). By clearing ECR[ETHER\_EN], configuration control registers such as the TCR and RCR are not reset, but the entire data path is reset.

**Table 40-92. ECR[ETHER\_EN] De-Assertion Effect on FEC**

| Register/Machine            | Reset Value                                |
|-----------------------------|--------------------------------------------|
| XMIT block                  | Transmission is aborted (bad CRC appended) |
| RECV block                  | Receive activity is aborted                |
| DMA block                   | All DMA activity is terminated             |
| RDAR                        | Cleared                                    |
| TDAR                        | Cleared                                    |
| Descriptor Controller block | Halt operation                             |

### 40.6.5 User initialization (Prior to Setting ECR[ETHER\_EN])

You need to initialize portions the FEC prior to setting the ECR[ETHER\_EN] bit. The exact values depend on the particular application. The sequence is not important.

The following registers require initialization:

- EIMR
- EIR (clear by writing FFFF\_FFFFh)
- TFWR (optional)
- IALR / IAUR
- GAUR / GALR
- PALR / PAUR (only needed for full duplex flow control)
- OPD (only needed for full duplex flow control)
- RCR
- TCR
- MSCR (optional)
- MIB\_RAM (clear)
- FRSR (optional)
- EMRBR
- ERDSR
- ETDSR
- Transmit Descriptor ring (empty)
- Receive Descriptor ring (empty)

### **40.6.6 Microcontroller initialization**

In the FEC, the descriptor control RISC initializes some registers after ECR[ETHER\_EN] is asserted. After the microcontroller initialization sequence is complete, hardware is ready for operation.

The sequence is as follows:

1. Initialize the backoff random number seed.
2. Activate receiver.
3. Activate transmitter.
4. Clear transmit FIFO.

5. Clear receive FIFO.
6. Initialize the transmit ring pointer.
7. Initialize the receive ring pointer.
8. Initialize FIFO count registers.

### 40.6.7 User initialization (after setting ECR[ETHER\_EN])

After setting ECR[ETHER\_EN], you can set up the buffer/frame descriptors and write to TDAR and RDAR. See [Buffer descriptors](#), for more details.

### 40.6.8 Network interface options

The FEC supports an MII and reduced MII interface for 10/100 Mbps Ethernet, as well as a 7-wire serial interface for 10 Mbps Ethernet. The RCR[MII\_MODE] and RCR[RMII\_MODE] bits select the interface mode. In MII mode (RCR[MII\_MODE]=1 and RCR[RMII\_MODE]=0), there are 18 signals defined by the IEEE 802.3 standard and supported by the EMAC. The following table shows these signals.

**Table 40-93. MII mode**

| Signal description           | EMAC pin     |
|------------------------------|--------------|
| Transmit Clock               | FEC_TXCLK    |
| Transmit Enable              | FEC_TXEN     |
| Transmit Data                | FEC_TXD[3:0] |
| Transmit Error               | FEC_TXER     |
| Collision                    | FEC_COL      |
| Carrier Sense                | FEC_CRS      |
| Receive Clock                | FEC_RXCLK    |
| Receive Data Valid           | FEC_RXDV     |
| Receive Data                 | FEC_RXD[3:0] |
| Receive Error                | FEC_RXER     |
| Management Data Clock        | FEC_MDC      |
| Management Data Input/Output | FEC_MDIO     |

In RMII mode (RCR[MII\_MODE]=1 and RCR[RMII\_MODE]=1), the EMAC supports 8 external signals. These signals are shown in the table below.

**Table 40-94. RMII mode configuration**

| Signal description | EMAC pin     |
|--------------------|--------------|
| Reference Clock    | FEC_TXCLK    |
| Transmit Enable    | FEC_TXEN     |
| Transmit Data      | FEC_TXD[1:0] |
| Receive Data Valid | FEC_RXDV     |
| Receive Data       | FEC_RXD[1:0] |
| Receive Error      | FEC_RXER     |

The 7-wire serial mode interface (RCR[MII\_MODE]=1 and RCR[RMII\_MODE]=0 or 1) is generally referred to as AMD mode. The following table shows the 7-wire mode connections to the external transceiver.

**Table 40-95. 7-Wire mode configuration**

| Signal description | EMAC pin   |
|--------------------|------------|
| Transmit Clock     | FEC_TXCLK  |
| Transmit Enable    | FEC_TXEN   |
| Transmit Data      | FEC_TXD[0] |
| Collision          | FEC_COL    |
| Receive Clock      | FEC_RXCLK  |
| Receive Data Valid | FEC_RXDV   |
| Receive Data       | FEC_RXD[0] |

### 40.6.9 FEC frame transmission

The Ethernet transmitter is designed to work with almost no intervention from software. After ECR[ETHER\_EN] is set and data appears in the transmit FIFO, the Ethernet MAC can transmit onto the network. The Ethernet controller transmits bytes least significant bit (lsb) first.

When the transmit FIFO fills to the watermark (defined by TFWR), MAC transmit logic asserts FEC\_TXEN and starts transmitting the (PA) sequence, the start frame delimiter (SFD), and then the frame information from the FIFO. However, the controller defers the transmission if the network is busy (FEC\_CRS is asserted). Before transmitting, the controller waits for carrier sense to become inactive, then determines if carrier sense stays inactive for 60 bit times. If so, transmission begins after waiting an additional 36 bit times (96 bit times after carrier sense originally became inactive). See "Transmission errors" for more details.

If a collision occurs during transmission of the frame (half duplex mode), the Ethernet controller follows the specified backoff procedures and attempts to retransmit the frame until the retry limit is reached. The transmit FIFO stores at least the first 64 bytes of the transmit frame, so they do not have to be retrieved from system memory in case of a collision. This improves bus utilization and latency in case immediate retransmission is necessary.

When all the frame data is transmitted, FCS (frame check sequence) or 32-bit cyclic redundancy check (CRC) bytes are appended if the TC bit is set in the transmit frame control word. If the ABC bit is set in the transmit frame control word, a bad CRC is appended to the frame data regardless of the TC bit value. Following the transmission of the CRC, the Ethernet controller writes the frame status information to the MIB block. Transmit logic automatically pads short frames (if the TC bit in the transmit buffer descriptor for the end of frame buffer is set).

Settings in the EIMR determine interrupts generated to the buffer (TXB) and frame (TXF).

The transmit error interrupts are HBERR, BABT, LC, RL, and UN. If the transmit frame length exceeds MAX\_FL bytes, BABT interrupt is asserted. However, the entire frame is transmitted (no truncation).

To pause transmission, set TCR[GTS] (graceful transmit stop). The FEC transmitter stops immediately if transmission is not in progress; otherwise, it continues transmission until the current frame finishes or terminates with a collision. After the transmitter has stopped, the GRA (graceful stop complete) interrupt is asserted. If TCR[GTS] is cleared, the FEC resumes transmission with the next frame.

#### 40.6.9.1 Duplicate frame transmission

The FEC fetches transmit buffer descriptors (TxBDs) and the corresponding transmit data continuously until the transmit FIFO is full. It does not determine whether the TxBD to be fetched is already being processed internally (as a result of a wrap). As the FEC nears the end of the transmission of one frame, it begins to DMA the data for the next frame. To remain one BD ahead of the DMA, it also fetches the TxBD for the next frame. It is possible that the FEC fetches from memory a BD that has already been processed but not yet written back (it is read a second time with the R bit remains set). In this case, the data is fetched and transmitted again.

Using at least three TxBDs fixes this problem for large frames, but not for small frames. To ensure correct operation for large or small frames, one of the following must be true:

- The FEC software driver ensures that there is always at least one TxBD with the ready bit cleared.
- Every frame uses more than one TxBD and every TxBD but the last is written back immediately after the data is fetched.
- The FEC software driver ensures a minimum frame size,  $n$ . The minimum number of TxBDs is then  $(\text{Tx FIFO Size} \div (n+4))$  rounded up to the nearest integer (though the result cannot be less than three). The default Tx FIFO size is 192 bytes; this size is programmable.

## 40.6.10 FEC frame reception

The FEC receiver works with almost no intervention from the host and can perform address recognition, CRC checking, short frame checking, and maximum frame length checking. The Ethernet controller receives serial data lsb first.

When the driver enables the FEC receiver by setting ECR[ETHER\_EN], it immediately starts processing receive frames. When FEC\_RXDV is asserted, the receiver first checks for a valid PA/SFD header. If the PA/SFD is valid, it is stripped and the receiver processes the frame. If a valid PA/SFD is not found, the frame is ignored.

In serial mode, the first 16 bit times of RX\_D0 following assertion of FEC\_RXDV are ignored. Following the first 16 bit times, the data sequence is checked for alternating 1/0s. If a 11 or 00 data sequence is detected during bit times 17 to 21, the remainder of the frame is ignored. After bit time 21, the data sequence is monitored for a valid SFD (11). If a 00 is detected, the frame is rejected. When a 11 is detected, the PA/SFD sequence is complete.

In MII mode, the receiver checks for at least one byte matching the SFD. Zero or more PA bytes may occur, but if a 00 bit sequence is detected prior to the SFD byte, the frame is ignored.

After the first 6 bytes of the frame are received, the FEC performs address recognition on the frame.

After a collision window (64 bytes) of data is received and if address recognition has not rejected the frame, the receive FIFO signals the frame is accepted and may be passed on to the DMA. If the frame is a runt (due to collision) or is rejected by address recognition, the receive FIFO is notified to reject the frame. Therefore, no collision fragments are presented to you except late collisions, which indicate serious LAN problems.



During reception, the Ethernet controller checks for various error conditions and after the entire frame is written into the FIFO, a 32-bit frame status word is written into the FIFO. This status word contains the M, BC, MC, LG, NO, CR, OV, and TR status bits, and the frame length. See "Reception errors" for more details.

Receive buffer (RXB) and frame interrupts (RFINT) may be generated if enabled by the EIMR register. A receive error interrupt is a babbling receiver error (BABR). Receive frames are not truncated if they exceed the max frame length (MAX\_FL); however, the BABR interrupt occurs and the LG bit in the receive buffer descriptor (RxBD) is set. See [Ethernet Receive Buffer Descriptor \(RxBD\)](#), for more details.

When the receive frame is complete, the FEC sets the L-bit in the RxBD, writes the other frame status bits into the RxBD, and clears the E-bit. The Ethernet controller next generates a maskable interrupt (RFINT bit in EIR, maskable by RFIEN bit in EIMR), indicating that a frame is received and is in memory. The Ethernet controller then waits for a new frame.

### 40.6.11 Ethernet address recognition

The FEC filters the received frames based on destination address (DA) type — individual (unicast), group (multicast), or broadcast (all-ones group address). The difference between an individual address and a group address is determined by the I/G bit in the destination address field. A flowchart for address recognition on received frames appears in the figures below.

Address recognition is accomplished through the use of the receive block and microcode running on the microcontroller. The flowchart shown in [Figure 40-84](#) illustrates the address recognition decisions made by the receive block, while [Figure 40-85](#) illustrates the decisions made by the microcontroller.

If the DA is a broadcast address and broadcast reject (RCR[BC\_REJ]) is cleared, then the frame is accepted unconditionally, as shown in [Figure 40-84](#). Otherwise, if the DA is not a broadcast address, then the microcontroller runs the address recognition subroutine, as shown in [Figure 40-85](#).

If the DA is a group (multicast) address and flow control is disabled, then the microcontroller performs a group hash table lookup using the 64-entry hash table programmed in GAUR and GALR. If a hash match occurs, the receiver accepts the frame.

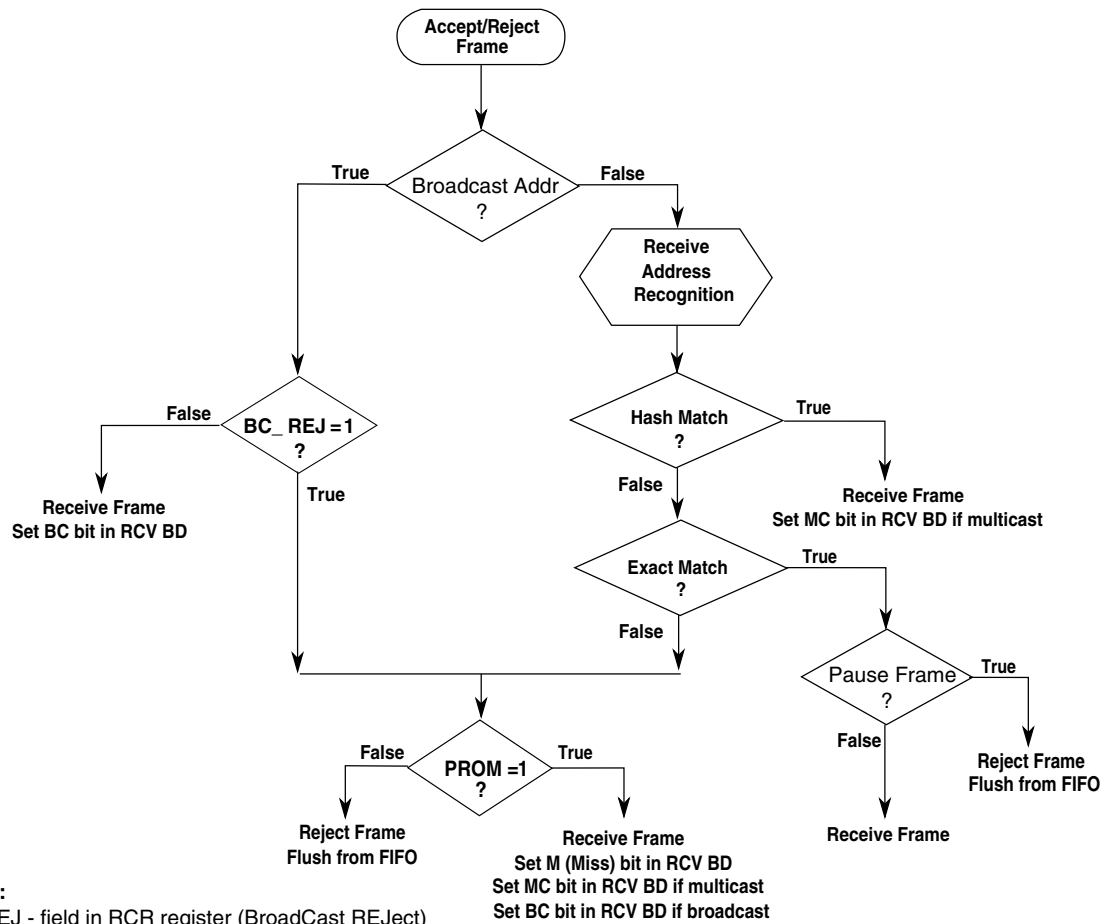
If flow control is enabled, the microcontroller does an exact address match check between the DA and the designated PAUSE DA (01:80:C2:00:00:01). If the receive block determines the received frame is a valid PAUSE frame, the frame is rejected. The receiver detects a PAUSE frame with the DA field set to the designated PAUSE DA or the unicast physical address.

If the DA is the individual (unicast) address, the microcontroller performs an individual exact match comparison between the DA and 48-bit physical address that you program in the PALR and PAUR registers. If an exact match occurs, the frame is accepted; otherwise, the microcontroller does an individual hash table lookup using the 64-entry hash table programmed in registers, IAUR and IALR. In the case of an individual hash match, the frame is accepted. Again, the receiver accepts or rejects the frame based on PAUSE frame detection, shown in [Figure 40-84](#).

If neither a hash match (group or individual) nor an exact match (group or individual) occur, and if promiscuous mode is enabled (RCR[PROM] set), the frame is accepted and the MISS bit in the receive buffer descriptor is set; otherwise, the frame is rejected.

Similarly, if the DA is a broadcast address, broadcast reject (RCR[BC\_REJ]) is asserted, and promiscuous mode is enabled, the frame is accepted and the MISS bit in the receive buffer descriptor is set; otherwise, the frame is rejected.

In general, when a frame is rejected, it is flushed from the FIFO.

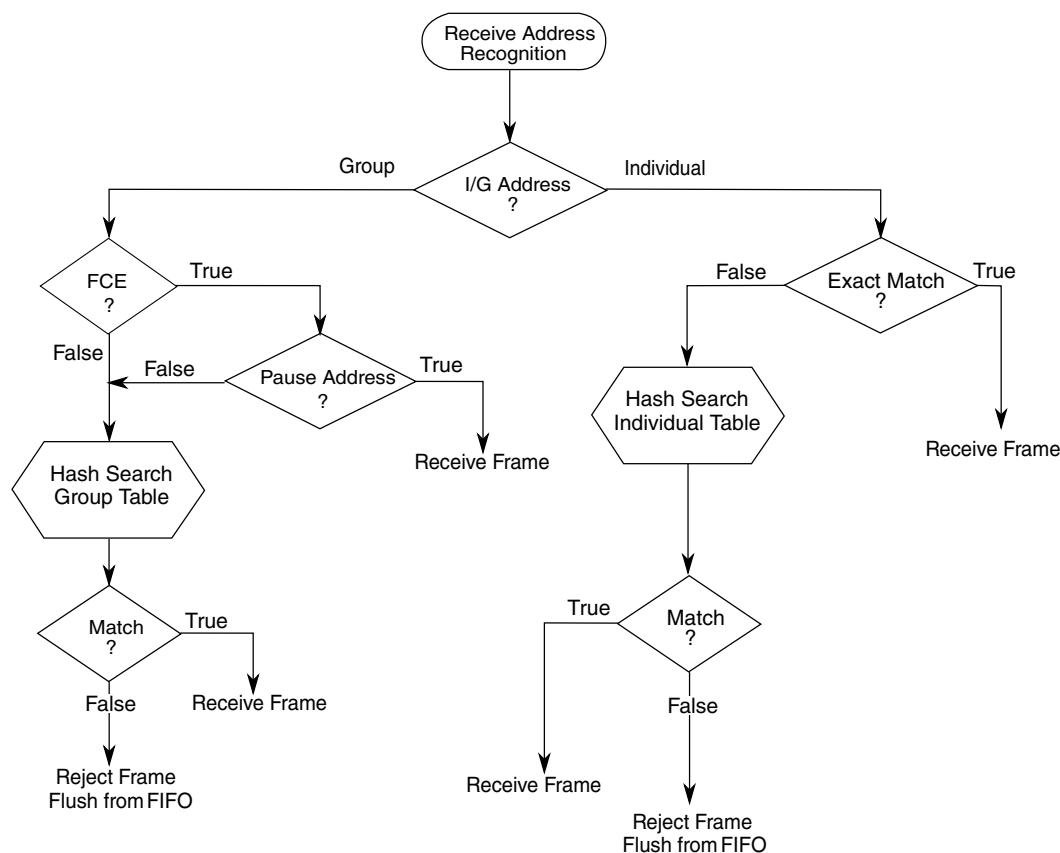
**Notes:**

BC\_REJ - field in RCR register (BroadCast REJect)

PROM - field in RCR register (PROMiscuous mode)

Pause Frame - valid PAUSE frame received

**Figure 40-84. Ethernet address recognition—receive block decisions**

**Notes:**

FCE - field in RCR register (flow control enable)

I/G - Individual/Group bit in destination address (lsb in first byte received in MAC frame)

**Figure 40-85. Ethernet address recognition—microcode decisions**

## 40.6.12 Hash algorithm

The hash table algorithm used in the group and individual hash filtering operates as follows. The 48-bit destination address is mapped into one of 64 bits, represented by 64 bits stored in GAUR, GALR (group address hash match), or IAUR, IALR (individual address hash match). This mapping is performed by passing the 48-bit address through the on-chip 32-bit CRC generator and selecting the six most significant bits of the CRC-encoded result to generate a number between 0 and 63. The msb of the CRC result selects GAUR (msb = 1) or GALR (msb = 0). The five least significant bits of the hash result select the bit within the selected register. If the CRC generator selects a bit set in the hash table, the frame is accepted; otherwise, it is rejected.

For example, if eight group addresses are stored in the hash table and random group addresses are received, the hash table prevents roughly 56/64 (87.5%) of the group address frames from reaching memory. Those that do reach memory must be further filtered by the processor to determine if they truly contain one of the eight desired addresses.

The effectiveness of the hash table declines as the number of addresses increases.

The user must initialize the hash table registers. Use this CRC32 polynomial to compute the hash:

$$X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$$

**Figure 40-86. CRC32 polynomial for hash computation**

The following table contains example destination addresses and corresponding hash values.

**Table 40-96. Destination address to 6-bit hash**

| 48-BIT DA      | 6-bit hash<br>(in hex) | Hash decimal value |
|----------------|------------------------|--------------------|
| 65FF_FFFF_FFFF | 0h                     | 0                  |
| 55FF_FFFF_FFFF | 1h                     | 1                  |
| 15FF_FFFF_FFFF | 2h                     | 2                  |
| 35FF_FFFF_FFFF | 3h                     | 3                  |
| B5FF_FFFF_FFFF | 4h                     | 4                  |
| 95FF_FFFF_FFFF | 5h                     | 5                  |
| D5FF_FFFF_FFFF | 6h                     | 6                  |
| F5FF_FFFF_FFFF | 7h                     | 7                  |
| DBFF_FFFF_FFFF | 8h                     | 8                  |
| FBFF_FFFF_FFFF | 9h                     | 9                  |
| BBFF_FFFF_FFFF | Ah                     | 10                 |
| 8BFF_FFFF_FFFF | Bh                     | 11                 |
| 0BFF_FFFF_FFFF | Ch                     | 12                 |
| 3BFF_FFFF_FFFF | Dh                     | 13                 |
| 7BFF_FFFF_FFFF | Eh                     | 14                 |
| 5BFF_FFFF_FFFF | Fh                     | 15                 |
| 27FF_FFFF_FFFF | 10h                    | 16                 |
| 07FF_FFFF_FFFF | 11h                    | 17                 |
| 57FF_FFFF_FFFF | 12h                    | 18                 |
| 77FF_FFFF_FFFF | 13h                    | 19                 |
| F7FF_FFFF_FFFF | 14h                    | 20                 |
| C7FF_FFFF_FFFF | 15h                    | 21                 |
| 97FF_FFFF_FFFF | 16h                    | 22                 |
| A7FF_FFFF_FFFF | 17h                    | 23                 |
| 99FF_FFFF_FFFF | 18h                    | 24                 |
| B9FF_FFFF_FFFF | 19h                    | 25                 |
| F9FF_FFFF_FFFF | 1Ah                    | 26                 |
| C9FF_FFFF_FFFF | 1Bh                    | 27                 |

*Table continues on the next page...*

**Table 40-96. Destination address to 6-bit hash (continued)**

| 48-BIT DA      | 6-bit hash<br>(in hex) | Hash decimal value |
|----------------|------------------------|--------------------|
| 59FF_FFFF_FFFF | 1Ch                    | 28                 |
| 79FF_FFFF_FFFF | 1Dh                    | 29                 |
| 29FF_FFFF_FFFF | 1Eh                    | 30                 |
| 19FF_FFFF_FFFF | 1Fh                    | 31                 |
| D1FF_FFFF_FFFF | 20h                    | 32                 |
| F1FF_FFFF_FFFF | 21h                    | 33                 |
| B1FF_FFFF_FFFF | 22h                    | 34                 |
| 91FF_FFFF_FFFF | 23h                    | 35                 |
| 11FF_FFFF_FFFF | 24h                    | 36                 |
| 31FF_FFFF_FFFF | 25h                    | 37                 |
| 71FF_FFFF_FFFF | 26h                    | 38                 |
| 51FF_FFFF_FFFF | 27h                    | 39                 |
| 7FFF_FFFF_FFFF | 28h                    | 40                 |
| 4FFF_FFFF_FFFF | 29h                    | 41                 |
| 1FFF_FFFF_FFFF | 2Ah                    | 42                 |
| 3FFF_FFFF_FFFF | 2Bh                    | 43                 |
| BFFF_FFFF_FFFF | 2Ch                    | 44                 |
| 9FFF_FFFF_FFFF | 2Dh                    | 45                 |
| DFFF_FFFF_FFFF | 2Eh                    | 46                 |
| EFFF_FFFF_FFFF | 2Fh                    | 47                 |
| 93FF_FFFF_FFFF | 30h                    | 48                 |
| B3FF_FFFF_FFFF | 31h                    | 49                 |
| F3FF_FFFF_FFFF | 32h                    | 50                 |
| D3FF_FFFF_FFFF | 33h                    | 51                 |
| 53FF_FFFF_FFFF | 34h                    | 52                 |
| 73FF_FFFF_FFFF | 35h                    | 53                 |
| 23FF_FFFF_FFFF | 36h                    | 54                 |
| 13FF_FFFF_FFFF | 37h                    | 55                 |
| 3DFF_FFFF_FFFF | 38h                    | 56                 |
| 0DFF_FFFF_FFFF | 39h                    | 57                 |
| 5DFF_FFFF_FFFF | 3Ah                    | 58                 |
| 7DFF_FFFF_FFFF | 3Bh                    | 59                 |
| FDFF_FFFF_FFFF | 3Ch                    | 60                 |
| DDFF_FFFF_FFFF | 3Dh                    | 61                 |
| 9DFF_FFFF_FFFF | 3Eh                    | 62                 |
| BDFF_FFFF_FFFF | 3Fh                    | 63                 |

### 40.6.13 Full duplex flow control

Full-duplex flow control allows you to transmit pause frames and to detect received pause frames. Upon detection of a pause frame, MAC data frame transmission stops for a given pause duration.

To enable PAUSE frame detection, the FEC must operate in full-duplex mode (TCR[FDEN] set) with flow control (RCR[FCE] set). The FEC detects a pause frame when the fields of the incoming frame match the pause frame specifications, as shown in the following table. In addition, the receive status associated with the frame should indicate that the frame is valid.

**Table 40-97. PAUSE frame field specification**

|                                   |                                     |
|-----------------------------------|-------------------------------------|
| <b>48-bit destination address</b> | 0180_C200_0001h or Physical Address |
| <b>48-bit source address</b>      | Any                                 |
| <b>16-bit type</b>                | 8808h                               |
| <b>16-bit opcode</b>              | 0001h                               |
| <b>16-bit PAUSE duration</b>      | 0000h — FFFFh                       |

The receiver and microcontroller modules perform PAUSE frame detection. The microcontroller runs an address recognition subroutine to detect the specified pause frame destination address, while the receiver detects the type and opcode pause frame fields. On detection of a pause frame, TCR[GTS] is set by the FEC internally. When transmission has paused, the EIR[GRA] interrupt is asserted and the pause timer begins to increment. The pause timer uses the transmit backoff timer hardware for tracking the appropriate collision backoff time in half-duplex mode. The pause timer increments once every slot time, until OPD[PAUSE\_DUR] slot times have expired. On OPD[PAUSE\_DUR] expiration, TCR[GTS] is cleared allowing MAC data frame transmission to resume. The receive flow control pause status bit (TCR[RFC\_PAUSE]) is set while the transmitter pauses due to reception of a pause frame.

To transmit a pause frame, the FEC must operate in full-duplex mode and you must set flow control pause (TCR[TFC\_PAUSE]). After TCR[TFC\_PAUSE] is set, the transmitter sets TCR[GTS] internally. When the transmission of data frames stops, the EIR[GRA] (graceful stop complete) interrupt asserts and the pause frame is transmitted. TCR[TFC\_PAUSE,GTS] are then cleared internally.

You must specify the desired pause duration in the OPD register.

When the transmitter pauses due to receiver/microcontroller pause frame detection, TCR[TFC\_PAUSE] may remain set and cause the transmission of a single pause frame. In this case, the EIR[GRA] interrupt is not asserted.

### 40.6.14 Inter-Packet Gap (IPG) time

The minimum inter-packet gap time for back-to-back transmission is 96 bit times. After completing a transmission or after the backoff algorithm completes, the transmitter waits for carrier sense to be negated before starting its 96 bit time IPG counter. Frame transmission may begin 96 bit times after carrier sense is negated if it stays negated for at least 60 bit times. If carrier sense asserts during the last 36 bit times, it is ignored and a collision occurs.

The receiver accepts back-to-back frames with a minimum spacing of at least 28 bit times. If an inter-packet gap between receive frames is less than 28 bit times, the receiver may discard the following frame.

### 40.6.15 Collision managing

If a collision occurs during frame transmission, the Ethernet controller continues the transmission for at least 32 bit times, transmitting a JAM pattern consisting of 32 ones. If the collision occurs during the sequence, a JAM pattern is sent after the end of the preamble sequence.

If a collision occurs within 512 bit times (one slot time), the retry process is initiated. The transmitter waits a random number of slot times. If a collision occurs after 512 bit times, then no retransmission is performed and the end of frame buffer is closed with a Late Collision (LC) error indication.

### 40.6.16 MII internal and external loopback

Internal and external loopback are supported by the Ethernet controller. In loopback mode, both of the FIFOs are used and the FEC actually operates in a full-duplex fashion. Internal and external loopback are configured using combinations of the RCR[LOOP, DRT] and TCR[FDEN] bits.

Set FDEN for internal and external loopback.

For internal loopback, set RCR[LOOP] and clear RCR[DRT]. FEC\_TXEN and FEC\_TXER do not assert during internal loopback. During internal loopback, the transmit/receive data rate is higher than in normal operation because the transmit and receive blocks use the internal bus clock instead of the clocks from the external transceiver. This causes an increase in the required system bus bandwidth for transmit



and receive data being DMA'd to/from external memory. It may be necessary to pace the frames on the transmit side and/or limit the size of the frames to prevent transmit FIFO underruns and receive FIFO overflows.

For external loopback, clear RCR[LOOP] and RCR[DRT], and configure the external transceiver for loopback.

### 40.6.17 RMII loopback

The FEC also supports RMII loopback. In this mode, transmit data is sent to the RMII receive logic and out the FEC\_TXD[1:0] pins. To enable RMII loopback mode, set RCR[RMII\_MODE, RMII\_LOOP] and TCR[FDEN] and clear RCR[RMII\_ECHO, LOOP].

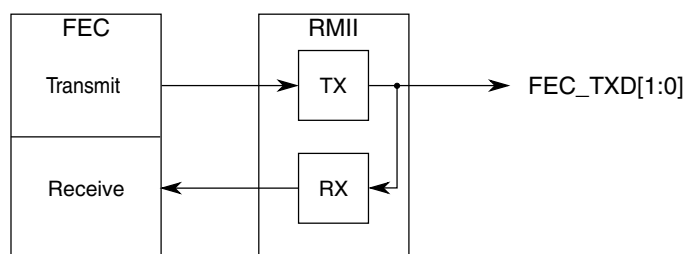


Figure 40-87. RMII loopback mode

### 40.6.18 RMII echo

The Ethernet controller also supports RMII echo mode, which transmits data on FEC\_TXD[1:0] as it is received on FEC\_RXD[1:0]. To enable RMII echo mode, set RCR[RMII\_MODE, RMII\_ECHO] bits and clear RCR[RMII\_LOOP, LOOP].

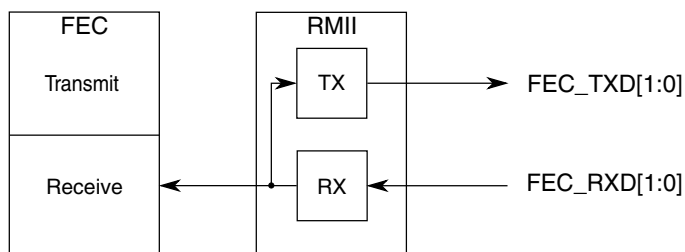


Figure 40-88. RMII echo mode

## 40.6.19 Ethernet error-managing procedure

The Ethernet controller reports frame reception and transmission error conditions using the MIB block counters, the FEC RxBDs, and the EIR register.

### 40.6.19.1 Transmission errors

#### 40.6.19.1.1 Transmitter underrun

If this error occurs, the FEC sends 32 bits that ensure a CRC error and stops transmitting. All remaining buffers for that frame are then flushed and closed, and EIR[UN] is set. The FEC then continues to the next transmit buffer descriptor and begin transmitting the next frame. The UN interrupt is asserted if enabled in the EIMR register.

#### 40.6.19.1.2 Retransmission attempts limit expired

When this error occurs, the FEC terminates transmission. All remaining buffers for that frame are flushed and closed, and EIR[RL] is set. The FEC then continues to the next transmit buffer descriptor and begins transmitting the next frame. The RL interrupt is asserted if enabled in the EIMR register.

#### 40.6.19.1.3 Late collision

When a collision occurs after the slot time (512 bits starting at the Preamble), the FEC terminates transmission. All remaining buffers for that frame are flushed and closed, and EIR[LC] is set. The FEC then continues to the next transmit buffer descriptor and begins transmitting the next frame. The LC interrupt is asserted if enabled in the EIMR register.

#### 40.6.19.1.4 Heartbeat

Some transceivers have a self-test feature called heartbeat or signal quality error. To signify a good self-test, the transceiver indicates a collision to the FEC within four microseconds after completion of a frame transmitted by the Ethernet controller. This indication of a collision does not imply a real collision error on the network, but is rather an indication that the transceiver continues to function properly. This is the heartbeat condition.

If TCR[HBC] is set and the heartbeat condition is not detected by the FEC after a frame transmission, a heartbeat error occurs. When this error occurs, the FEC closes the buffer, sets EIR[HB], and generates the HBERR interrupt if it is enabled.

## 40.6.19.2 Reception errors

### 40.6.19.2.1 Overrun error

If the receive block has data to put into the receive FIFO and the receive FIFO is full, FEC sets RxBD[OV]. All subsequent data in the frame is discarded and subsequent frames may also be discarded until the receive FIFO is serviced by the DMA and space is made available. At this point the receive frame/status word is written into the FIFO with the OV bit set. The driver must discard this frame.

### 40.6.19.2.2 Non-octet error (dribbling bits)

The Ethernet controller manages up to seven dribbling bits when the receive frame terminates past a non-octet aligned boundary. Dribbling bits are not used in the CRC calculation. If there is a CRC error, the frame non-octet aligned (NO) error is reported in the RxBD. If there is no CRC error, no error is reported.

### 40.6.19.2.3 CRC error

When a CRC error occurs with no dribble bits, FEC closes the buffer and sets RxBD[CR]. CRC checking cannot be disabled, but the CRC error can be ignored if checking is not required.

### 40.6.19.2.4 Frame length violation

When the receive frame length exceeds MAX\_FL bytes the BABR interrupt is generated, and RxBD[LG] is set. The frame is not truncated unless the frame length exceeds 2047 bytes.

### 40.6.19.2.5 Truncation

When the receive frame length exceeds 2047 bytes, frame is truncated and RxBD[TR] is set.



# Chapter 41

## SENT Receiver (SRX)

### 41.1 Chip-specific SRX information

This chip has two instances of the SRX module.

**Table 41-1. SRX module instances**

| Instance | Input/receive channels | FIFO depth |
|----------|------------------------|------------|
| SRX_0    | 6                      | 16         |
| SRX_1    | 6                      | 16         |

### 41.2 Introduction

The Single Edge Nibble Transmission (SENT) Receiver (SRX) module is a multi-channel receiver for receiving serial data frames which are being transmitted by a sensor implementing the SENT encoding scheme and present them to the CPU for further processing.

This module is based on the SAE J2716 information report titled "SENT - Single Edge Nibble Transmission for Automotive Applications" and released on January 27, 2010 (<http://www.sae.org>), Apr2007 and Feb2008. As per this standard, the SENT protocol is intended for use in applications where high resolution sensor data needs to be communicated from a sensor to an Engine Control Unit (ECU). It is intended as a replacement for the lower resolution methods of 10 bit A/Ds and PWM and as a simpler low-cost alternative to CAN or LIN. The implementation assumes that the sensor is a smart sensor containing a microprocessor or dedicated logic device (ASIC) to create the signal.

The SENT encoding scheme is a unidirectional communications scheme from the sensor / transmitting device to the controller /receiving device which does not include a coordination signal from the controller/receiving device. The sensor signal is transmitted as a series of pulses with data encoded as falling to falling edge periods.

### 41.2.1 Features

The SENT receiver module has the following features:

- The number of SENT channels supported are device-specific. See the device configuration details.
- Unified message read logic to transfer all messages received to system memory via interrupts or DMA
  - Programmable option per channel to support reading of messages via DMA or Interrupt
  - Internal round robin arbitration sequencer to loop across channels to read messages from each channel leaving CPU/DMA transparent to this process
  - A FIFO for storing the received Fast Messages from all channels that are to be read out via DMA. FIFO depth is device-specific. See the device configuration section for details.
  - All received messages are stored excluding the synchronization pulse (for Fast messages) and identifiers (for Slow Serial Messages)
  - Separate buffers for storing Fast and Slow Serial Messages
  - Separate DMA and Interrupts generated for Fast and Slow Serial Messages
- Supports compensation for variation in SENT Tx Clock up to  $\pm 25\%$  and adjusts its measurement for drift in the Transmitter and Receiver clocks per channel
  - Calibration pulse correction factor per channel is available for use by software
- Implements for each channel all receiver diagnostics as specified by the SAE Specifications
- Internal 4-bit CRC calculator that can be configured to compute CRC in both Legacy and Recommended method (as given in SAE SENT Specification) for both Fast and Short Serial Message per channel
- Internal 6-bit CRC calculator for Enhanced Serial Messages per channel

- Supports all bit rates by having per channel configurable Rx clock periods from 3  $\mu$ s to 90  $\mu$ s
- Support for configurable number of data nibbles per channel
- Option to disable individual channel via programmable control bit
- Time stamping on start of each valid message (i.e. without errors) across all channels. This time stamp is appended to each message for synchronization of messages and for application to understand how much time has elapsed between successive sensor readings
- Support for pause pulse per channel. This can be enabled by programming a bit in the control register for that channel
- Input Programmable Filter on each channel to filter any glitches on input. This filter duration is programmable by the user software.
- 32-bit register interface for programming the module's registers and reading the received messages
- Works on two clocks, one for accurate receive operations (high frequency receiver clock or protocol clock) and other for message read logic (system bus clock) and the module can work irrespective of the frequency relationship between the two clocks. See [Clocks and resets](#) for more details on the clocks.

## 41.2.2 Modes of operation

Modes of operation.

### 41.2.2.1 Debug mode

In debug mode, message reception will be halted (without waiting for the current message being received to be completed) and all existing status and messages will be retained. DBG\_FRZ bit is required to be set for this mode. On debug exit, a calibration length error (CAL\_LEN\_ERR) can be reported and the channels will start receiving messages from a valid calibration pulse.

### 41.2.2.2 Low power modes

SENT Receiver behavior is affected by chip-specific low power modes (for example, STOP or WAIT modes). These low power modes should either gate-off both the system bus clock and the protocol clocks or reduce their frequencies, depending on the application.

On exiting from these low power modes, calibration length error (CAL\_LEN\_ERR) can get set in channel status registers as the SENT sensor (transmitter) could be in the middle of a message transmission at that point. Thus, software should clear all error status bits after exiting the low power modes.

**Modes where both system bus clock and protocol clock are gated:** In this case, both message reception and interrupt / DMA request generation will halt. The internal receiver channels will freeze (since clock is not there) and will not receive any messages. Messages which are completely received before entering the low power mode can be read after the low power mode exit.

**Modes where frequency of system bus clock is changed:** For reduced power consumption the system bus clock can be reduced but not below a minimum value (See [System bus clock requirements](#)) to avoid overflow of received messages. Protocol clock should not go below the minimum value (See [High frequency receiver clock \(protocol clock\) requirements](#)) for accurate receive operations.

#### Note

Gating of clocks in the low power mode is chip-specific, see the chapter that refers how the modules behave in low power modes.

#### NOTE

If the SOC requests entry to debug or stop mode, the message being received currently is discarded and the SOC enters debug/stop mode immediately. This does not affect the messages received completely prior to entering debug mode and will still be present on the message buffer and registers until they are read out.

#### NOTE

The message read registers [Channel 'n' Fast Message Data Read Register (n = 0 to (CH-1)) (CHn\_FMSG\_DATA), Channel 'n' Fast Message CRC Read Register (n = 0 to (CH-1)) (CHn\_FMSG\_CRC), Channel 'n' Fast Message Time Stamp Read Register (n = 0 to (CH-1)) (CHn\_FMSG\_TS), Channel 'n' Serial Message Read Register (Bit 3) (n = 0 to (CH-1))



(CHn\_SMSG\_BIT3), Channel 'n' Serial Message Read Register (Bit 2) (n = 0 to (CH-1)) (CHn\_SMSG\_BIT2), Channel 'n' Serial Message Time Stamp Read Register (n = 0 to (CH-1)) (CHn\_SMSG\_TS), DMA Fast Message Data Read Register (DMA\_FMSG\_DATA), DMA Fast Message CRC Read Register (DMA\_FMSG\_CRC), DMA Fast Message Time Stamp Read Register (DMA\_FMSG\_TS), DMA Slow Serial Message Bit3 Read Register (DMA\_SMSG\_BIT3), DMA Slow Serial Message Bit2 Read Register (DMA\_SMSG\_BIT2) and DMA Slow Serial Message Time Stamp Read Register (DMA\_SMSG\_TS)] will appear to lose their contents in the following conditions:

- The very first message is being received but not yet completely received and the MCU enters debug or freeze mode, the current message reception will get discarded and message read registers (as mentioned above) will read zeros
- Auto clear functionality is enabled (GBL\_CTRL[FAST\_CLR] = 1). In this case, when first message is read, it will get clear the message read registers (due to auto clear functionality being enabled). On reading again, the message read registers (as mentioned above) might read zeros.

If the MCU requests entry to debug or stop mode, the message being received currently by SENT Receiver is discarded and the MCU enters debug/stop mode immediately. This does not affect the messages received completely, prior to entering debug mode and these messages will still be present on the message buffer and registers until they are read out. Thus allow one message to be received completely and do not enable "Auto Clear" (GBL\_CTRL[FAST\_CLR] = 0), to allow messages to be read in debug or stop mode.

## 41.2.3 Block diagram

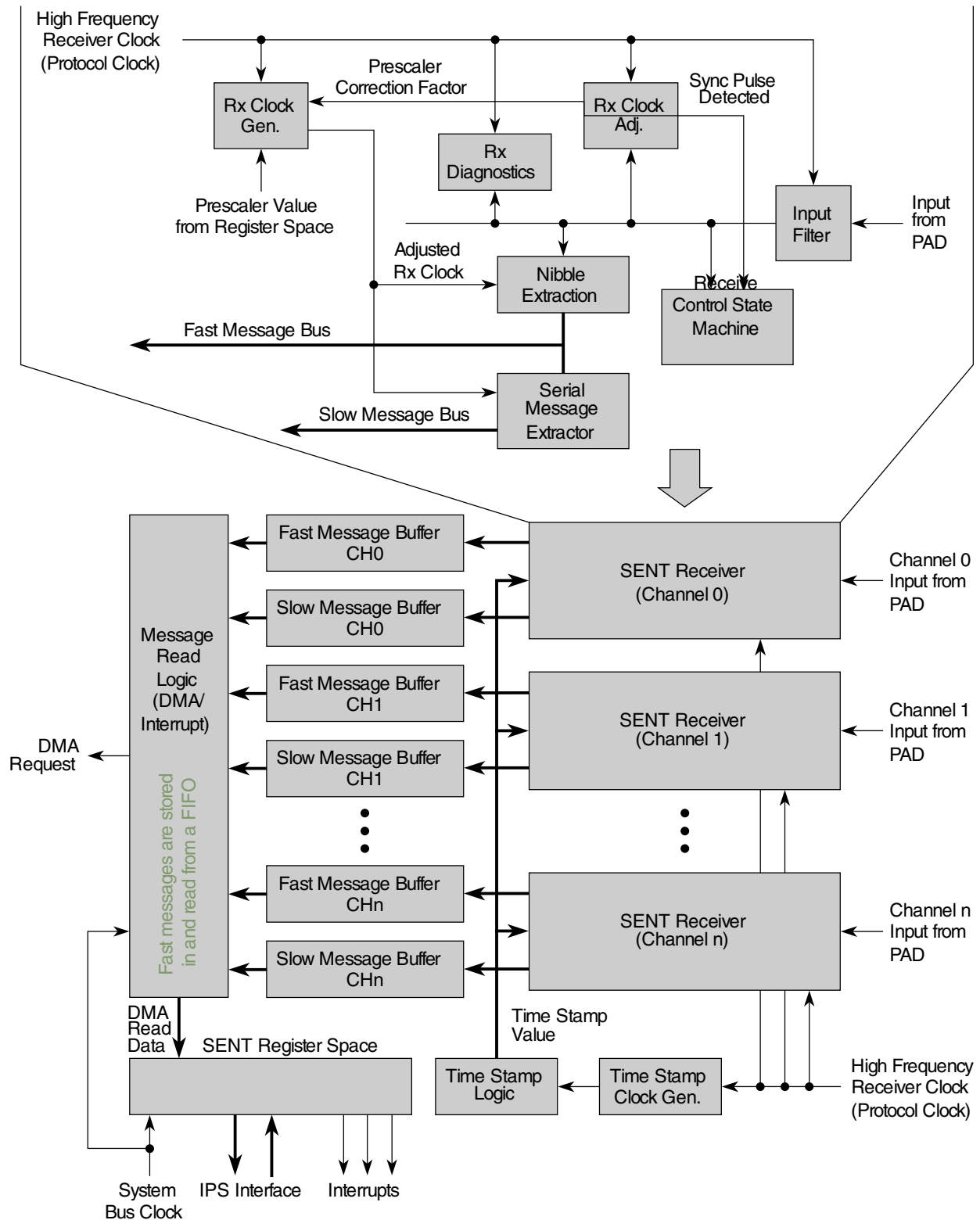


**Figure 41-1. SENT Receiver block diagram**

## 41.2.4 Design overview

The SENT Receiver Module is a unified receiver module comprising of a number of receiver channels and a unified DMA and Register Space for accessing the messages from each channel. [Figure 41-1](#) shows the top level block diagram for the SENT Receiver module.

Each channel of the SENT Receiver is composed of a Clock Generation and Adjustment block to generate the receiver clock and adjust it for frequency variation in the transmitter clock. This block also detects the synchronization/calibration pulse which is present in every SENT message.

The SENT Receiver State Machine controls the sequencing of decoding the messages and combining the information as nibbles. In parallel a diagnostic block is running that monitors the incoming messages for any errors and also verifies the CRC of the received messages. The fast and serial message extractors collate the message nibbles and add the channel number and time stamp information to the messages. A free running time stamp counter is used to add the time stamp value to each message received. The value of this counter is input to each channel and the channel logic appends the time stamp to each message. These messages are then formatted to be stored in the DMA FIFOs (for channels which are selected to be connected to the DMA) or in corresponding message read buffer. Only the messages that pass the diagnostics checks are forwarded to be stored in the FIFOs or buffers.

The SENT Receiver provides two mechanisms for reading the received messages, which are via DMA transfers and via Interrupts. Based on the control bit settings in the control registers, a DMA request or Interrupt is asserted to signal the DMA controller or CPU that messages are available to be read.

The free running time stamp counter, register block and DMA interface are common for all channels configured. The register space contains all the necessary registers for all channels. The register space is 32-bit aligned and accessible via 8-/16-/32-bit accesses. The registers for all un-configured channels are marked as reserved.

### 41.2.4.1 Functional overview

After reset, the SENT module comes up in the disabled state. All channel enable bits and the global enable bit are set to 0 and the user software must program the SENT module parameters correctly before enabling it. The user software must correctly configure all parameters to avoid any underflow or overflow in the message FIFOs or buffers.

Once enabled, the Receiver Clock Generation and Adjustment block detects the synchronization pulse. Until then, the state machine blocks all subsequent operations of the receiver. Once the synchronization pulse is detected, the receiver clock tick period is compensated for variation in transmitter clock tick period and the state machine then waits for the reception of data nibbles. The adjusted receiver clock is used to sample the data nibbles from the incoming serial message. The state machine waits until the status and communication nibble and configured number of data nibbles are received. The state machine then waits for the pause pulse (if configured to receive a pause pulse). The fast and serial message extractor block provides the state machine with the information on how many nibbles have been received. All diagnostics checks are computed in parallel. Once all required data nibbles are received, the CRC is checked, time stamp and channel number is added to the message. The time stamp for a Fast Message is sampled at the falling edge that comes at the end of the calibration pulse (which is also the start of the status and communication nibble pulse) and is stored in the message read buffer along with the message. This final message is stored in the DMA FIFO or message buffer as described in the subsequent sections. Any error detected during the receive process causes the status bits in the corresponding channel's status register to be set.

The above process repeats across each enabled channel in parallel and keeps pushing error free messages into the FIFOs or buffers on high frequency receiver clock. The module's message read logic synchronizes all these messages on the bus clock and pops out these messages on any DMA or CPU read access. Any DMA underflow errors are stored in the global status register while the overflow errors are stored in the corresponding channel's status register. In case of an overflow in Fast Messages, the newer messages received will overwrite the previous message.

For detailed description on the complete functionality of each block, see [Functional description](#).

## 41.3 External signal description

This section lists all inputs to the SENT Receiver block.

### 41.3.1 SENT\_RX [(CH-1):0]

These pins serve as the serial data input for each of the SENT Receiver channel. There is one bit per channel supported by the device. The SENT sensor input pads on the device using this module should have pull-up enabled to ensure a 1'b1 is seen at the module input when SENT sensor is not driving any data.

### Note

The value of CH is device-specific. See the device configuration details.

## 41.4 Memory map and register definition

This section provides a detailed description of all registers accessible in the SENT Receiver block.

The following table gives an overview on all SENT Receiver registers. The memory map comprises of 32-bit aligned registers which can be accessed via 8-bit, 16-bit, or 32-bit accesses. Write access to reserved locations will generate transfer error. Read access to reserved locations will also generate transfer error and the read data bus will show all 0s. The Memory Map and complete module is in Big Endian Format.

The module will not check for correctness of programmed values in Register Space and software must ensure that correct values are being written.

### NOTE

Register Address = Base Address + Address Offset, where the Base Address is defined at the MCU level and the Address Offset is defined at the module level. The number of registers are device specific. See the device configuration section for details.

### NOTE

The value of the CH is device-specific. See the device configuration section for details.

### SRX memory map

| Address offset (hex) | Register name                                            | Width (in bits) | Access | Reset value | Section/ page               |
|----------------------|----------------------------------------------------------|-----------------|--------|-------------|-----------------------------|
| 0                    | Global Control Register (SRX_GBL_CTRL)                   | 32              | R/W    | 0001_0010h  | <a href="#">41.4.1/2053</a> |
| 4                    | Channel Enable Register (SRX_CHNL_EN)                    | 32              | R/W    | 0000_0000h  | <a href="#">41.4.2/2055</a> |
| 8                    | Global Status Register (SRX_GBL_STATUS)                  | 32              | w1c    | 0000_0000h  | <a href="#">41.4.3/2056</a> |
| C                    | Fast Message Ready Status Register (SRX_FMSG_RDY)        | 32              | R/W    | 0000_0000h  | <a href="#">41.4.4/2057</a> |
| 10                   | Slow Serial Message Ready Status Register (SRX_SMSG_RDY) | 32              | R/W    | 0000_0000h  | <a href="#">41.4.5/2058</a> |
| 18                   | Data Control Register 1 (SRX_DATA_CTRL1)                 | 32              | R/W    | 1111_1100h  | <a href="#">41.4.6/2059</a> |

*Table continues on the next page...*

## SRX memory map (continued)

| Address offset (hex) | Register name                                                           | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|-------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 28                   | Fast Message DMA Control Register (SRX_FDMA_CTRL)                       | 32              | R/W    | 0000_0000h  | <a href="#">41.4.7/2061</a>  |
| 2C                   | Slow Serial Message DMA Control Register (SRX_SDMA_CTRL)                | 32              | R/W    | 0000_0000h  | <a href="#">41.4.8/2061</a>  |
| 34                   | Fast Message Ready Interrupt Control Register (SRX_FRDY_IE)             | 32              | R/W    | 0000_0000h  | <a href="#">41.4.9/2062</a>  |
| 38                   | Slow Serial Message Ready Interrupt Enable Register (SRX_SRDY_IE)       | 32              | R/W    | 0000_0000h  | <a href="#">41.4.10/2063</a> |
| 40                   | DMA Fast Message Data Read Register (SRX_DMA_FMSG_DATA)                 | 32              | R      | 0000_0000h  | <a href="#">41.4.11/2063</a> |
| 44                   | DMA Fast Message CRC Read Register (SRX_DMA_FMSG_CRC)                   | 32              | R      | 0000_0000h  | <a href="#">41.4.12/2064</a> |
| 48                   | DMA Fast Message Time Stamp Read Register (SRX_DMA_FMSG_TS)             | 32              | R      | 0000_0000h  | <a href="#">41.4.13/2065</a> |
| 50                   | DMA Slow Serial Message Bit3 Read Register (SRX_DMA_SMSG_BIT3)          | 32              | R      | 0000_0000h  | <a href="#">41.4.14/2065</a> |
| 54                   | DMA Slow Serial Message Bit2 Read Register (SRX_DMA_SMSG_BIT2)          | 32              | R      | 0000_0000h  | <a href="#">41.4.15/2067</a> |
| 58                   | DMA Slow Serial Message Time Stamp Read Register (SRX_DMA_SMSG_TS)      | 32              | R      | 0000_0000h  | <a href="#">41.4.16/2067</a> |
| 60                   | Channel 'n' Clock Control Register (n = 0 to (CH-1)) (SRX_CH0_CLK_CTRL) | 32              | R/W    | 0000_8000h  | <a href="#">41.4.17/2068</a> |
| 64                   | Channel 'n' Status Register (n=0 to (CH-1)) (SRX_CH0_STATUS)            | 32              | w1c    | 0000_0000h  | <a href="#">41.4.18/2070</a> |
| 68                   | Channel 'n' Configuration Register (n=0 to (CH-1)) (SRX_CH0_CONFIG)     | 32              | R/W    | 0000_0104h  | <a href="#">41.4.19/2073</a> |
| 70                   | Channel 'n' Clock Control Register (n = 0 to (CH-1)) (SRX_CH1_CLK_CTRL) | 32              | R/W    | 0000_8000h  | <a href="#">41.4.17/2068</a> |
| 74                   | Channel 'n' Status Register (n=0 to (CH-1)) (SRX_CH1_STATUS)            | 32              | w1c    | 0000_0000h  | <a href="#">41.4.18/2070</a> |
| 78                   | Channel 'n' Configuration Register (n=0 to (CH-1)) (SRX_CH1_CONFIG)     | 32              | R/W    | 0000_0104h  | <a href="#">41.4.19/2073</a> |
| 80                   | Channel 'n' Clock Control Register (n = 0 to (CH-1)) (SRX_CH2_CLK_CTRL) | 32              | R/W    | 0000_8000h  | <a href="#">41.4.17/2068</a> |
| 84                   | Channel 'n' Status Register (n=0 to (CH-1)) (SRX_CH2_STATUS)            | 32              | w1c    | 0000_0000h  | <a href="#">41.4.18/2070</a> |
| 88                   | Channel 'n' Configuration Register (n=0 to (CH-1)) (SRX_CH2_CONFIG)     | 32              | R/W    | 0000_0104h  | <a href="#">41.4.19/2073</a> |
| 90                   | Channel 'n' Clock Control Register (n = 0 to (CH-1)) (SRX_CH3_CLK_CTRL) | 32              | R/W    | 0000_8000h  | <a href="#">41.4.17/2068</a> |
| 94                   | Channel 'n' Status Register (n=0 to (CH-1)) (SRX_CH3_STATUS)            | 32              | w1c    | 0000_0000h  | <a href="#">41.4.18/2070</a> |
| 98                   | Channel 'n' Configuration Register (n=0 to (CH-1)) (SRX_CH3_CONFIG)     | 32              | R/W    | 0000_0104h  | <a href="#">41.4.19/2073</a> |
| A0                   | Channel 'n' Clock Control Register (n = 0 to (CH-1)) (SRX_CH4_CLK_CTRL) | 32              | R/W    | 0000_8000h  | <a href="#">41.4.17/2068</a> |

Table continues on the next page...

## SRX memory map (continued)

| Address offset (hex) | Register name                                                                         | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|---------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| A4                   | Channel 'n' Status Register (n=0 to (CH-1)) (SRX_CH4_STATUS)                          | 32              | w1c    | 0000_0000h  | <a href="#">41.4.18/2070</a> |
| A8                   | Channel 'n' Configuration Register (n=0 to (CH-1)) (SRX_CH4_CONFIG)                   | 32              | R/W    | 0000_0104h  | <a href="#">41.4.19/2073</a> |
| B0                   | Channel 'n' Clock Control Register (n = 0 to (CH-1)) (SRX_CH5_CLK_CTRL)               | 32              | R/W    | 0000_8000h  | <a href="#">41.4.17/2068</a> |
| B4                   | Channel 'n' Status Register (n=0 to (CH-1)) (SRX_CH5_STATUS)                          | 32              | w1c    | 0000_0000h  | <a href="#">41.4.18/2070</a> |
| B8                   | Channel 'n' Configuration Register (n=0 to (CH-1)) (SRX_CH5_CONFIG)                   | 32              | R/W    | 0000_0104h  | <a href="#">41.4.19/2073</a> |
| 160                  | Channel 'n' Fast Message Data Read Register (n=0 to (CH-1)) (SRX_CH0_FMSG_DATA)       | 32              | R      | 0000_0000h  | <a href="#">41.4.20/2076</a> |
| 164                  | Channel 'n' Fast Message CRC Read Register (n=0 to (CH-1)) (SRX_CH0_FMSG_CRC)         | 32              | R      | 0000_0000h  | <a href="#">41.4.21/2077</a> |
| 168                  | Channel 'n' Fast Message Time Stamp Read Register (n=0 to (CH-1)) (SRX_CH0_FMSG_TS)   | 32              | R      | 0000_0000h  | <a href="#">41.4.22/2077</a> |
| 16C                  | Channel 'n' Serial Message Read Register (Bit 3) (n=0 to (CH-1)) (SRX_CH0_SMSG_BIT3)  | 32              | R      | 0000_0000h  | <a href="#">41.4.23/2078</a> |
| 170                  | Channel 'n' Serial Message Read Register (Bit 2)(n=0 to (CH-1)) (SRX_CH0_SMSG_BIT2)   | 32              | R      | 0000_0000h  | <a href="#">41.4.24/2079</a> |
| 174                  | Channel 'n' Serial Message Time Stamp Read Register (n=0 to (CH-1)) (SRX_CH0_SMSG_TS) | 32              | R      | 0000_0000h  | <a href="#">41.4.25/2080</a> |
| 178                  | Channel 'n' Fast Message Data Read Register (n=0 to (CH-1)) (SRX_CH1_FMSG_DATA)       | 32              | R      | 0000_0000h  | <a href="#">41.4.20/2076</a> |
| 17C                  | Channel 'n' Fast Message CRC Read Register (n=0 to (CH-1)) (SRX_CH1_FMSG_CRC)         | 32              | R      | 0000_0000h  | <a href="#">41.4.21/2077</a> |
| 180                  | Channel 'n' Fast Message Time Stamp Read Register (n=0 to (CH-1)) (SRX_CH1_FMSG_TS)   | 32              | R      | 0000_0000h  | <a href="#">41.4.22/2077</a> |
| 184                  | Channel 'n' Serial Message Read Register (Bit 3) (n=0 to (CH-1)) (SRX_CH1_SMSG_BIT3)  | 32              | R      | 0000_0000h  | <a href="#">41.4.23/2078</a> |
| 188                  | Channel 'n' Serial Message Read Register (Bit 2)(n=0 to (CH-1)) (SRX_CH1_SMSG_BIT2)   | 32              | R      | 0000_0000h  | <a href="#">41.4.24/2079</a> |
| 18C                  | Channel 'n' Serial Message Time Stamp Read Register (n=0 to (CH-1)) (SRX_CH1_SMSG_TS) | 32              | R      | 0000_0000h  | <a href="#">41.4.25/2080</a> |
| 190                  | Channel 'n' Fast Message Data Read Register (n=0 to (CH-1)) (SRX_CH2_FMSG_DATA)       | 32              | R      | 0000_0000h  | <a href="#">41.4.20/2076</a> |
| 194                  | Channel 'n' Fast Message CRC Read Register (n=0 to (CH-1)) (SRX_CH2_FMSG_CRC)         | 32              | R      | 0000_0000h  | <a href="#">41.4.21/2077</a> |
| 198                  | Channel 'n' Fast Message Time Stamp Read Register (n=0 to (CH-1)) (SRX_CH2_FMSG_TS)   | 32              | R      | 0000_0000h  | <a href="#">41.4.22/2077</a> |
| 19C                  | Channel 'n' Serial Message Read Register (Bit 3) (n=0 to (CH-1)) (SRX_CH2_SMSG_BIT3)  | 32              | R      | 0000_0000h  | <a href="#">41.4.23/2078</a> |
| 1A0                  | Channel 'n' Serial Message Read Register (Bit 2)(n=0 to (CH-1)) (SRX_CH2_SMSG_BIT2)   | 32              | R      | 0000_0000h  | <a href="#">41.4.24/2079</a> |

Table continues on the next page...



## SRX memory map (continued)

| Address offset (hex) | Register name                                                                         | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|---------------------------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 1A4                  | Channel 'n' Serial Message Time Stamp Read Register (n=0 to (CH-1)) (SRX_CH2_SMSG_TS) | 32              | R      | 0000_0000h  | <a href="#">41.4.25/2080</a> |
| 1A8                  | Channel 'n' Fast Message Data Read Register (n=0 to (CH-1)) (SRX_CH3_FMSG_DATA)       | 32              | R      | 0000_0000h  | <a href="#">41.4.20/2076</a> |
| 1AC                  | Channel 'n' Fast Message CRC Read Register (n=0 to (CH-1)) (SRX_CH3_FMSG_CRC)         | 32              | R      | 0000_0000h  | <a href="#">41.4.21/2077</a> |
| 1B0                  | Channel 'n' Fast Message Time Stamp Read Register (n=0 to (CH-1)) (SRX_CH3_FMSG_TS)   | 32              | R      | 0000_0000h  | <a href="#">41.4.22/2077</a> |
| 1B4                  | Channel 'n' Serial Message Read Register (Bit 3) (n=0 to (CH-1)) (SRX_CH3_SMSG_BIT3)  | 32              | R      | 0000_0000h  | <a href="#">41.4.23/2078</a> |
| 1B8                  | Channel 'n' Serial Message Read Register (Bit 2)(n=0 to (CH-1)) (SRX_CH3_SMSG_BIT2)   | 32              | R      | 0000_0000h  | <a href="#">41.4.24/2079</a> |
| 1BC                  | Channel 'n' Serial Message Time Stamp Read Register (n=0 to (CH-1)) (SRX_CH3_SMSG_TS) | 32              | R      | 0000_0000h  | <a href="#">41.4.25/2080</a> |
| 1C0                  | Channel 'n' Fast Message Data Read Register (n=0 to (CH-1)) (SRX_CH4_FMSG_DATA)       | 32              | R      | 0000_0000h  | <a href="#">41.4.20/2076</a> |
| 1C4                  | Channel 'n' Fast Message CRC Read Register (n=0 to (CH-1)) (SRX_CH4_FMSG_CRC)         | 32              | R      | 0000_0000h  | <a href="#">41.4.21/2077</a> |
| 1C8                  | Channel 'n' Fast Message Time Stamp Read Register (n=0 to (CH-1)) (SRX_CH4_FMSG_TS)   | 32              | R      | 0000_0000h  | <a href="#">41.4.22/2077</a> |
| 1CC                  | Channel 'n' Serial Message Read Register (Bit 3) (n=0 to (CH-1)) (SRX_CH4_SMSG_BIT3)  | 32              | R      | 0000_0000h  | <a href="#">41.4.23/2078</a> |
| 1D0                  | Channel 'n' Serial Message Read Register (Bit 2)(n=0 to (CH-1)) (SRX_CH4_SMSG_BIT2)   | 32              | R      | 0000_0000h  | <a href="#">41.4.24/2079</a> |
| 1D4                  | Channel 'n' Serial Message Time Stamp Read Register (n=0 to (CH-1)) (SRX_CH4_SMSG_TS) | 32              | R      | 0000_0000h  | <a href="#">41.4.25/2080</a> |
| 1D8                  | Channel 'n' Fast Message Data Read Register (n=0 to (CH-1)) (SRX_CH5_FMSG_DATA)       | 32              | R      | 0000_0000h  | <a href="#">41.4.20/2076</a> |
| 1DC                  | Channel 'n' Fast Message CRC Read Register (n=0 to (CH-1)) (SRX_CH5_FMSG_CRC)         | 32              | R      | 0000_0000h  | <a href="#">41.4.21/2077</a> |
| 1E0                  | Channel 'n' Fast Message Time Stamp Read Register (n=0 to (CH-1)) (SRX_CH5_FMSG_TS)   | 32              | R      | 0000_0000h  | <a href="#">41.4.22/2077</a> |
| 1E4                  | Channel 'n' Serial Message Read Register (Bit 3) (n=0 to (CH-1)) (SRX_CH5_SMSG_BIT3)  | 32              | R      | 0000_0000h  | <a href="#">41.4.23/2078</a> |
| 1E8                  | Channel 'n' Serial Message Read Register (Bit 2)(n=0 to (CH-1)) (SRX_CH5_SMSG_BIT2)   | 32              | R      | 0000_0000h  | <a href="#">41.4.24/2079</a> |
| 1EC                  | Channel 'n' Serial Message Time Stamp Read Register (n=0 to (CH-1)) (SRX_CH5_SMSG_TS) | 32              | R      | 0000_0000h  | <a href="#">41.4.25/2080</a> |



### 41.4.1 Global Control Register (SRX\_GBL\_CTRL)

This register provides the global control and setting for the SENT Receiver Module. The SENT\_EN bit must be programmed after all other modules settings and control bits have been written to.

Address: 0h base + 0h offset = 0h

|       |                   |    |    |    |    |        |        |        |        |         |    |          |    |         |    |         |
|-------|-------------------|----|----|----|----|--------|--------|--------|--------|---------|----|----------|----|---------|----|---------|
| Bit   | 0                 | 1  | 2  | 3  | 4  | 5      | 6      | 7      | 8      | 9       | 10 | 11       | 12 | 13      | 14 | 15      |
| R     | TSPRSC            |    |    |    |    |        |        | 0      | FIFOWM |         |    |          |    |         |    |         |
| W     |                   |    |    |    |    |        |        |        |        |         |    |          |    |         |    |         |
| Reset | 0                 | 0  | 0  | 0  | 0  | 0      | 0      | 0      | 0      | 0       | 0  | 0        | 0  | 0       | 0  | 1       |
| Bit   | 16                | 17 | 18 | 19 | 20 | 21     | 22     | 23     | 24     | 25      | 26 | 27       | 28 | 29      | 30 | 31      |
| R     | NIB_LEN_VAR_LIMIT | 0  |    |    |    | FMFOIE | FMDUIE | SMDUIE | 0      | FIFO_EN | 0  | FAST_CLR | 0  | DBG_FRZ | 0  | SENT_EN |
| W     |                   |    |    |    |    |        |        |        |        |         |    |          |    |         |    |         |
| Reset | 0                 | 0  | 0  | 0  | 0  | 0      | 0      | 0      | 0      | 0       | 0  | 1        | 0  | 0       | 0  | 0       |

**SRX\_GBL\_CTRL field descriptions**

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–7<br>TSPRSC    | Time Stamp Prescaler Value. Value in number of clock cycles of high frequency receiver clock set in order to obtain a clock frequency for the time stamp counter block.<br><br>0 to 255 - Time Stamp Clock = (High Frequency Clock)/(Value + 1). Value set by user software should be such that the clock period is at least 1 $\mu$ s.<br><br>If the high frequency clock input is 75 MHz, then the value set in this field would be 0x4A (or 74) to obtain a 1 $\mu$ s clock.<br><br>Note: Since the time stamp clock is common for all channels, the timestamp clock frequency must be at least twice the fastest receiver channel in the application. This is to ensure that the time stamp clock is fast enough to show change in timestamp from message to message. |
| 8–10<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 11–15<br>FIFOWM  | DMA Read FIFO Water Mark Level. Indicates the number of Fast Messages to be stored in FIFO before a DMA request is asserted. Default = 1 message.<br><br>0 Invalid Value. Should not be used<br>1 to 31 Valid water mark levels. Value fixed by user software.<br><br><b>NOTE:</b> The number of bytes to be transferred set in the DMA Controller's TCD must be set equal to the number of bytes corresponding to the set water mark value.                                                                                                                                                                                                                                                                                                                              |

*Table continues on the next page...*

## SRX\_GBL\_CTRL field descriptions (continued)

| Field                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 16<br>NIB_LEN_VAR_LIMIT | Decides the allowable jitter/clock drift limit for nibble length measurement.<br><br><b>NOTE:</b> This bit should be set to 1 for proper functioning of Sent Rx.<br><br>0 Allows a jitter/clock drift of approx upto 12.5% of transmitter clock tick period. This actual allowed jitter/drift value is dependent on the transmitter clock tick period and HF receiver clock period.<br>1 Allows a jitter/clock drift as per SAE spec (50%).                |
| 17–20<br>Reserved       | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                    |
| 21<br>FMFOIE            | Fast Message FIFO Overflow Interrupt Enable: Enables interrupt assertion when Fast Message is received on a channel which has DMA enabled after FIFO is full. Default value is 0.<br><br>0 Interrupt is disabled<br>1 Interrupt is enabled                                                                                                                                                                                                                 |
| 22<br>FMDUIE            | Fast Message DMA Underflow Interrupt Enable: Enables interrupt assertion when there is an underflow condition on Fast Message DMA registers. Default value is 0.<br><br>0 Interrupt is disabled<br>1 Interrupt is enabled                                                                                                                                                                                                                                  |
| 23<br>SMDUIE            | Slow Serial Message DMA Underflow Interrupt Enable: Enables interrupt assertion when there is an underflow condition on Slow Message DMA registers. Default value is 0.<br><br>0 Interrupt is disabled<br>1 Interrupt is enabled                                                                                                                                                                                                                           |
| 24<br>Reserved          | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                    |
| 25<br>FIFO_EN           | FIFO enable bit. Enables FIFO when DMA is enabled. When this bit is low, data is read from module as explained in <a href="#">DMA request for Fast Message reading when FIFO is disabled</a> and when the bit is high, data is read from module as explained in <a href="#">DMA request for Fast Message reading via FIFO</a> . Default value is 0.<br><br>0 FIFO is disabled<br>1 FIFO is enabled                                                         |
| 26<br>Reserved          | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                    |
| 27<br>FAST_CLR          | Fast Clearing Enable bit. Enables clearing of ready status (for both fast and slow) bit automatically when corresponding message is read. Default value is 1. See <a href="#">Fast Message Ready Status Register (SRX_FMSG_RDY)</a> and <a href="#">Slow Serial Message Ready Status Register (SRX_SMSG_RDY)</a> for details on the ready status bits.<br><br>0 Fast Clearing is disabled. Bits will be cleared by writing 1<br>1 Fast Clearing is enabled |
| 28<br>Reserved          | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                    |
| 29<br>DBG_FRZ           | Debug Freeze. This bit will enable the debug mode support. SENT Module will freeze in debug mode if this bit is set. Default is 0.<br><br>0 No effect in debug mode<br>1 Freeze in debug mode                                                                                                                                                                                                                                                              |

Table continues on the next page...

**SRX\_GBL\_CTRL field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                                                              |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 30<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                  |
| 31<br>SENT_EN  | SENT Receiver Global Enable. This bit enables or disables the complete receiver module irrespective of the setting of individual channel enable bits. User software must program all channel parameters before setting this bit. Default is 0.<br><br>0 Entire module is disabled<br>1 Module is enabled |

**41.4.2 Channel Enable Register (SRX\_CHNL\_EN)**

This register is used to enable individual SENT Receiver channels in the module. The user software must ensure all channel parameters and control settings have been programmed before writing to this bit. In case these parameters are changed while the channel is enabled, normal operation is not guaranteed.

**NOTE**

Reads of bits beyond those for the supported number of channels should be ignored, and these bits must be written to 0.

Address: 0h base + 4h offset = 4h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25    | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | EN_CH |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0     | 0  | 0  | 0  | 0  | 0  | 0  |

**SRX\_CHNL\_EN field descriptions**

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                            |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–25<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                |
| 26–31<br>EN_CH   | Enable bits for channels 5 to 0 . All channels are disabled after reset. User software must program all channel parameters before setting this bit. Once set, other channel parameters in registers should not be changed until this bit is cleared. Only Interrupt Enable and Status bits can be written to.<br><br>0 Channel is disabled<br>1 Channel is enabled. Channel parameter registers are locked for writing |

41.4.3 Global Status Register (SRX\_GBL\_STATUS)

This register provides the status of various conditions in the module that are common across all channels.

Address: 0h base + 8h offset = 8h

|       |    |    |    |    |    |      |      |      |    |    |    |    |    |    |    |    |
|-------|----|----|----|----|----|------|------|------|----|----|----|----|----|----|----|----|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5    | 6    | 7    | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0  |    |    |    |    |      |      |      |    |    |    |    |    |    |    |    |
| W     |    |    |    |    |    |      |      |      |    |    |    |    |    |    |    |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0    | 0    | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21   | 22   | 23   | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0  |    |    |    |    | FMFO | FMDU | SMDU | 0  |    |    |    |    |    |    |    |
| W     |    |    |    |    |    | w1c  | w1c  | w1c  |    |    |    |    |    |    |    |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0    | 0    | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

SRX\_GBL\_STATUS field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–20<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                         |
| 21<br>FMFO       | Fast Message FIFO Overflow: Interrupt status bit which indicates that Fast Message is received on any channel which has DMA enabled after FIFO is full. Default value is 0. This bit is valid only when fifo is enabled<br><br>0 No Overflow<br>1 Overflow occurred                                                                                                                                                                                                                             |
| 22<br>FMDU       | Fast Message DMA Underflow: Interrupt Status bit that indicates when there is an underflow in Fast Message buffers or FIFO in any channel which has DMA enabled. If FIFO is enabled, it is set when FIFO is empty and software still reads Fast Message DMA registers. And if FIFO is not enabled, it is set when there is no message in the DMA buffers for read and the software still reads the Fast Message DMA register. Default value is 0.<br><br>0 No underflow<br>1 Underflow occurred |
| 23<br>SMDU       | Slow Serial Message DMA Underflow: Interrupt Status bit that indicates when there is an Underflow in Slow Serial Message DMA buffers on channel which has DMA enabled. The bit is set when there is no                                                                                                                                                                                                                                                                                          |

Table continues on the next page...

**SRX\_GBL\_STATUS field descriptions (continued)**

| Field             | Description                                                                                                                                                           |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | message in the DMA buffers for read and the software still reads the Slow Serial Message registers. Default value is 0.<br><br>0 No underflow<br>1 Underflow occurred |
| 24–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                               |

**41.4.4 Fast Message Ready Status Register (SRX\_FMSG\_RDY)**

The bits of this register are asserted when a Fast Message arrives on a corresponding channel that is not configured to be read via DMA. If a channel is not configured for DMA (via [Fast Message DMA Control Register \(SRX\\_FDMA\\_CTRL\)](#) ), it is assumed to be read via Interrupt or Polling this status bit. Even if the Interrupt Enable bit for that channel is not set (via [Fast Message Ready Interrupt Control Register \(SRX\\_FRDY\\_IE\)](#) ) and the channel is enabled, the message ready status bit in this register will be asserted on Fast Message reception. Software must ensure that if a channel is enabled; either DMA or Interrupt must be enabled for that channel. Proper flow cannot be guaranteed if neither is selected.

These bits can be configured to be cleared via "Write 1 to Clear" or "Fast Clearing Mechanism". The configuration is done by writing to the FAST\_CLR bit in the [Global Control Register \(SRX\\_GBL\\_CTRL\)](#) .

**NOTE**

Reads of bits beyond those for the supported number of channels should be ignored. Writes to these bits is ignored by the Receiver.

Address: 0h base + Ch offset = Ch

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25    | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | F_RDY |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0     | 0  | 0  | 0  | 0  | 0  | 0  |

**SRX\_FMSG\_RDY field descriptions**

| Field            | Description                                                                                                                                                                                                        |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–25<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                            |
| 26–31<br>F_RDY   | Fast Message Data Ready Status bits for channels 5 to 0 . Each bit indicates that a Fast Message is ready to be read by CPU from that specific channel. This bit is asserted only when DMA is NOT enabled for that |

*Table continues on the next page...*

**SRX\_FMSG\_RDY field descriptions (continued)**

| Field | Description                                                                                                                                                                                                                                                                                                                          |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | channel (from the DMA Control Register) and an error free message is received, irrespective of the value of corresponding enable bit in Message Ready Interrupt Control Register. Reading the message from the corresponding memory-mapped Fast Message Read Registers would automatically clear this bit (Fast Clearing Mechanism). |
|       | <b>NOTE:</b> These bits are either "write 1 to clear" (w1c) or self-clearing depending on the value of the FAST_CLR BIT in the Global Control Register (GBL_CTRL).                                                                                                                                                                   |
|       | The clearing mechanism is selected by user by programming the FAST_CLR bit in the Global Control register. See Section <a href="#">Global Control Register (SRX_GBL_CTRL)</a> for details on FAST_CLR bit.                                                                                                                           |
|       | <b>NOTE:</b> When using the fast clearing option, these bits will not clear till the complete message is read out which requires the CPU to read all 3 fast message registers via 32-bit read accesses. For 8/16-bit accesses fast clearing mechanism should not be used as it is not supported.                                     |
|       | <b>NOTE:</b> When the fast clearing option is not configured, the bits will be cleared only if 1 is written to those bits.                                                                                                                                                                                                           |
|       | 0 No Fast Message is available to be read out via Interrupt                                                                                                                                                                                                                                                                          |
|       | 1 Fast Message is available to be read out via Interrupt                                                                                                                                                                                                                                                                             |

#### 41.4.5 Slow Serial Message Ready Status Register (SRX\_SMSG\_RDY)

The bits of this register are asserted when a Slow Serial Message arrives on a corresponding channel that is not configured to be read via DMA. If a channel is not configured for DMA (via [Fast Message DMA Control Register \(SRX\\_FDMA\\_CTRL\)](#) and [Slow Serial Message DMA Control Register \(SRX\\_SDMA\\_CTRL\)](#)), it is assumed to be read via Interrupt or Polling this status bit. Even if the Interrupt Enable bit for that channel is not set (via [Fast Message Ready Interrupt Control Register \(SRX\\_FRDY\\_IE\)](#) and [Slow Serial Message Ready Interrupt Enable Register \(SRX\\_SRDY\\_IE\)](#)) and the channel is enabled, the message ready status bit in this register will be asserted on Slow Message reception. Software must ensure that if a channel is enabled; either DMA or Interrupt must be enabled for that channel. Erroneous flow can happen if neither is selected.

These bits can be configured to be cleared via "Write 1 to Clear" or "Fast Clearing Mechanism". The configuration is done by writing to the FAST\_CLR bit in the [Global Control Register \(SRX\\_GBL\\_CTRL\)](#).

#### NOTE

Reads of bits beyond those for the supported number of channels should be ignored. Writes to these bits is ignored by the Receiver.

Address: 0h base + 10h offset = 10h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25    | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | S_RDY |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0     | 0  | 0  | 0  | 0  | 0  | 0  |

**SRX\_SMSG\_RDY field descriptions**

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–25<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 26–31<br>S_RDY   | <p>Slow Serial Message Data Ready Status bits for channels 5 to 0. Each bit indicates that a Slow Serial Message is ready to be read by CPU from that specific channel. This bit is asserted only when DMA is NOT enabled for that channel (from the DMA Control Register) and an error free message is received, irrespective of the value of corresponding enable bit in Message Ready Interrupt Control Register. Reading the message from the corresponding memory-mapped Serial Message Read Registers would automatically clear this bit (Fast Clearing Mechanism).</p> <p><b>NOTE:</b> These bits are either "write 1 to clear" (w1c) or self-clearing depending on the value of the FAST_CLR bit in the Global Control Register (GBL_CTRL).</p> <p>The clearing mechanism is selected by user by programming the FAST_CLR bit in the Global Control register. See Section <a href="#">Global Control Register (SRX_GBL_CTRL)</a> for details on FAST_CLR bit.</p> <p><b>NOTE:</b> When using the fast clearing option, these bits will not clear till the complete message is read out which requires the CPU to read all 3 Fast Message registers via 32-bit read accesses. For 8/16-bit accesses fast clearing mechanism should not be used as it is not supported.</p> <p><b>NOTE:</b> When the fast clearing option is not configured, the bits will be cleared only if 1 is written to those bits.</p> <p>0 No Slow Serial Message is available to be read out via Interrupt<br/>1 Slow Serial Message is available to be read out via Interrupt</p> |

**41.4.6 Data Control Register 1 (SRX\_DATA\_CTRL1)**

The Data Control Register 1 (DATA\_CTRL1) is described below.

**NOTE**

Reset value changes depending on the number of channels that are enabled on the device.

Address: 0h base + 18h offset = 18h

|       |   |         |   |   |   |         |   |   |   |         |    |    |    |         |    |    |
|-------|---|---------|---|---|---|---------|---|---|---|---------|----|----|----|---------|----|----|
| Bit   | 0 | 1       | 2 | 3 | 4 | 5       | 6 | 7 | 8 | 9       | 10 | 11 | 12 | 13      | 14 | 15 |
| R     | 0 | NIBBCH0 |   |   | 0 | NIBBCH1 |   |   | 0 | NIBBCH2 |    |    | 0  | NIBBCH3 |    |    |
| W     |   |         |   |   |   |         |   |   |   |         |    |    |    |         |    |    |
| Reset | 0 | 0       | 0 | 1 | 0 | 0       | 0 | 1 | 0 | 0       | 0  | 1  | 0  | 0       | 0  | 1  |

|       |    |         |    |    |    |         |    |    |    |          |    |    |    |          |    |    |
|-------|----|---------|----|----|----|---------|----|----|----|----------|----|----|----|----------|----|----|
| Bit   | 16 | 17      | 18 | 19 | 20 | 21      | 22 | 23 | 24 | 25       | 26 | 27 | 28 | 29       | 30 | 31 |
| R     | 0  | NIBBCH4 |    |    | 0  | NIBBCH5 |    |    | 0  | Reserved |    |    | 0  | Reserved |    |    |
| W     |    |         |    |    |    |         |    |    |    |          |    |    |    |          |    |    |
| Reset | 0  | 0       | 0  | 1  | 0  | 0       | 0  | 1  | 0  | 0        | 0  | 0  | 0  | 0        | 0  | 0  |

**SRX\_DATA\_CTRL1 field descriptions**

| Field             | Description                                                                                                                                                     |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>Reserved     | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                         |
| 1–3<br>NIBBCH0    | Number of Data Nibbles supported in Channel 0<br>0 and 7 Invalid value. Must not be used<br>1 to 6 Number of data nibbles to be supported by respective channel |
| 4<br>Reserved     | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                         |
| 5–7<br>NIBBCH1    | Number of Data Nibbles supported in Channel 1<br>0 and 7 Invalid value. Must not be used<br>1 to 6 Number of data nibbles to be supported by respective channel |
| 8<br>Reserved     | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                         |
| 9–11<br>NIBBCH2   | Number of Data Nibbles supported in Channel 2<br>0 and 7 Invalid value. Must not be used<br>1 to 6 Number of data nibbles to be supported by respective channel |
| 12<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                         |
| 13–15<br>NIBBCH3  | Number of Data Nibbles supported in Channel 3<br>0 and 7 Invalid value. Must not be used<br>1 to 6 Number of data nibbles to be supported by respective channel |
| 16<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                         |
| 17–19<br>NIBBCH4  | Number of Data Nibbles supported in Channel 4<br>0 and 7 Invalid value. Must not be used<br>1 to 6 Number of data nibbles to be supported by respective channel |
| 20<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                         |
| 21–23<br>NIBBCH5  | Number of Data Nibbles supported in Channel 5<br>0 and 7 Invalid value. Must not be used<br>1 to 6 Number of data nibbles to be supported by respective channel |
| 24<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                         |
| 25–27<br>Reserved | This field is reserved.                                                                                                                                         |
| 28<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                         |
| 29–31<br>Reserved | This field is reserved.                                                                                                                                         |



### 41.4.7 Fast Message DMA Control Register (SRX\_FDMA\_CTRL)

#### NOTE

Reads of bits beyond those for the supported number of channels should be ignored. Writes to these bits is ignored by the Receiver.

RW

Address: 0h base + 28h offset = 28h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25      | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | FDMA_EN |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0       | 0  | 0  | 0  | 0  | 0  | 0  |

#### SRX\_FDMA\_CTRL field descriptions

| Field            | Description                                                                                                                                                                                                                          |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–25<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                              |
| 26–31<br>FDMA_EN | Enable DMA for Fast Messages on channels 5 to 0 . These bits are writable when corresponding Fast Message Ready Interrupt Enable bits are set to 0.<br><br>0 DMA for Fast Messages is disabled<br>1 DMA for Fast Messages is enabled |

### 41.4.8 Slow Serial Message DMA Control Register (SRX\_SDMA\_CTRL)

#### NOTE

Reads of bits beyond those for the supported number of channels should be ignored. Writes to these bits is ignored by the Receiver.

Address: 0h base + 2Ch offset = 2Ch

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25      | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | SDMA_EN |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0       | 0  | 0  | 0  | 0  | 0  | 0  |

**SRX\_SDMA\_CTRL field descriptions**

| Field            | Description                                                                                                                                                                                                                                                      |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–25<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                          |
| 26–31<br>SDMA_EN | Enable DMA for Slow Serial Messages on channels 5 to 0 . These bits are writable when corresponding Slow Serial Message Ready Interrupt Enable bits are set to 0.<br><br>0 DMA for Slow Serial Messages is disabled<br>1 DMA for Slow Serial Messages is enabled |

**41.4.9 Fast Message Ready Interrupt Control Register (SRX\_FRDY\_IE)****NOTE**

Reads of bits beyond those for the supported number of channels should be ignored. Writes to these bits is ignored by the Receiver.

Address: 0h base + 34h offset = 34h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25      | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | FRDY_IE |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0       | 0  | 0  | 0  | 0  | 0  | 0  |

**SRX\_FRDY\_IE field descriptions**

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–25<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 26–31<br>FRDY_IE | Enable for Fast Message Ready Interrupt on channels 5 to 0 . These bits are writable when corresponding DMA enable bits are set to 0 in the Fast Message DMA Control Register. The availability of message is indicated via assertion of corresponding bit in Fast Message Ready Status Register irrespective of the value set in corresponding interrupt enable bit and DMA is not enabled. When a bit is asserted in this register, an interrupt for corresponding channel is generated.<br><br>0 Interrupt on reception of Fast Messages is disabled<br>1 Interrupt on reception of Fast Messages is enabled |

### 41.4.10 Slow Serial Message Ready Interrupt Enable Register (SRX\_SRDY\_IE)

#### NOTE

Reads of bits beyond those for the supported number of channels should be ignored. Writes to these bits is ignored by the Receiver.

Address: 0h base + 38h offset = 38h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25      | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | SRDY_IE |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0       | 0  | 0  | 0  | 0  | 0  | 0  |

#### SRX\_SRDY\_IE field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–25<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 26–31<br>SRDY_IE | Enable for Slow Serial Message Ready Interrupt on channels 5 to 0 . These bits are writable when corresponding DMA enable bits are set to 0 in the Slow Serial DMA Control Register. The availability of message is indicated via assertion of corresponding bit in Slow Serial Message Ready Status Register irrespective of the value set in corresponding interrupt enable bit and DMA is not enabled. When a bit in this register is asserted, an interrupt for corresponding channel is generated.<br><br>0 Interrupt on reception of Slow Serial Messages is disabled<br>1 Interrupt on reception of Slow Serial Messages is enabled |

### 41.4.11 DMA Fast Message Data Read Register (SRX\_DMA\_FMSG\_DATA)

DMA Source Size must be set to 32 bits when reading this register (DMA\_FMSG\_DATA) via DMA. When FIFO is enabled, the DMA must read the complete message using this register only. Three read accesses must be made to read complete message.

Address: 0h base + 40h offset = 40h

|       |       |   |   |   |       |   |   |   |       |   |    |    |       |    |    |    |       |    |    |    |       |    |    |    |       |    |    |    |       |    |    |    |
|-------|-------|---|---|---|-------|---|---|---|-------|---|----|----|-------|----|----|----|-------|----|----|----|-------|----|----|----|-------|----|----|----|-------|----|----|----|
| Bit   | 0     | 1 | 2 | 3 | 4     | 5 | 6 | 7 | 8     | 9 | 10 | 11 | 12    | 13 | 14 | 15 | 16    | 17 | 18 | 19 | 20    | 21 | 22 | 23 | 24    | 25 | 26 | 27 | 28    | 29 | 30 | 31 |
| R     | CHNUM |   |   |   | SCNIB |   |   |   | DNIB1 |   |    |    | DNIB2 |    |    |    | DNIB3 |    |    |    | DNIB4 |    |    |    | DNIB5 |    |    |    | DNIB6 |    |    |    |
| W     |       |   |   |   |       |   |   |   |       |   |    |    |       |    |    |    |       |    |    |    |       |    |    |    |       |    |    |    |       |    |    |    |
| Reset | 0     | 0 | 0 | 0 | 0     | 0 | 0 | 0 | 0     | 0 | 0  | 0  | 0     | 0  | 0  | 0  | 0     | 0  | 0  | 0  | 0     | 0  | 0  | 0  | 0     | 0  | 0  | 0  | 0     | 0  | 0  | 0  |

**SRX\_DMA\_FMSG\_DATA field descriptions**

| Field          | Description                                                                                                       |
|----------------|-------------------------------------------------------------------------------------------------------------------|
| 0–3<br>CHNUM   | Channel Number. Indicates the channel number of the received message being read by DMA. Valid values are 0 to 5 . |
| 4–7<br>SCNIB   | Status and Communication Nibble of message                                                                        |
| 8–11<br>DNIB1  | Data Nibble 1. Always supported                                                                                   |
| 12–15<br>DNIB2 | Data Nibble 2. Should be ignored if number of data nibbles supported by channel is 1                              |
| 16–19<br>DNIB3 | Data Nibble 3. Should be ignored if number of data nibbles supported by channel is 2 or less                      |
| 20–23<br>DNIB4 | Data Nibble 4. Should be ignored if number of data nibbles supported by channel is 3 or less                      |
| 24–27<br>DNIB5 | Data Nibble 5. Should be ignored if number of data nibbles supported by channel is 4 or less                      |
| 28–31<br>DNIB6 | Data Nibble 6. Should be ignored if number of data nibbles supported by channel is 5 or less                      |

#### 41.4.12 DMA Fast Message CRC Read Register (SRX\_DMA\_FMSG\_CRC)

DMA Source Size must be set to 32 bits when reading this register (DMA\_FMSG\_CRC) via DMA. When FIFO is enabled, fast messages are read out from FIFO through the DMA Fast Message Data Read register only. Reading this register will generate an access error.

Address: 0h base + 44h offset = 44h

|       |   |   |   |   |   |   |   |   |   |   |    |    |       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12    | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    | CRC4b |    |    |    | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0     | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

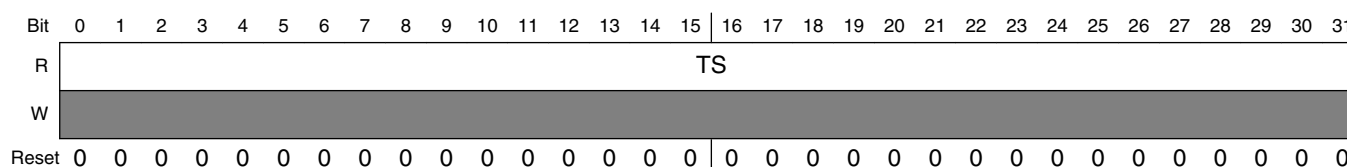
**SRX\_DMA\_FMSG\_CRC field descriptions**

| Field             | Description                                                                             |
|-------------------|-----------------------------------------------------------------------------------------|
| 0–11<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 12–15<br>CRC4b    | 4-bit CRC value of the message.                                                         |
| 16–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |

### 41.4.13 DMA Fast Message Time Stamp Read Register (SRX\_DMA\_FMSG\_TS)

The resolution of the time stamp is set by programming the Time Stamp Prescaler Value in the [Global Control Register \(SRX\\_GBL\\_CTRL\)](#) register. DMA Source Size must be set to 32 bits when reading this register (DMA\_FMSG\_TS) via DMA. When FIFO is enabled, fast messages are read out from FIFO through the DMA Fast Message Data Read register only. Reading this register will generate an access error.

Address: 0h base + 48h offset = 48h



**SRX\_DMA\_FMSG\_TS field descriptions**

| Field      | Description                                                                                                                                                                                                           |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>TS | Time Stamp for received message. Indicates the relative time when the falling edge that comes at the end of the calibration pulse (which is also the start of the status and communication nibble pulse) was detected |

### 41.4.14 DMA Slow Serial Message Bit3 Read Register (SRX\_DMA\_SMSG\_BIT3)

The above register (DMA\_SMSG\_BIT3) is common for all types of slow serial messages. Bit C defines the type of enhanced packet and bit TYPE defines whether it is short serial or enhanced serial message. Since this register is used to read both Short and Enhanced Serial Messages, the register fields have been placed in such a way to match the bit positions in actual message defined by SAE Specification as shown in [Table 41-83](#), [Table 41-84](#) and [Table 41-85](#). [Table 41-83](#) and [Table 41-84](#) shows the enhanced serial messages format to be used when reading this register for these two types of messages. [Table 41-85](#) shows the short serial message.

DMA Source Size must be set to 32 bits when reading this register via DMA.

#### NOTE

If the TYPE bit in the register reads '0', then remainder bits (for CFG, ID and DATA) will be redundant and read zeros. The short serial message will be located in DMA\_SMSG\_BIT2 register fields. If TYPE bit is '1', then the message in register is

Enhanced Serial Message and using CFG bit, software can decode the ID and DATA fields.

Address: 0h base + 50h offset = 50h

|       |       |   |   |   |      |   |   |   |   |   |    |    |    |    |    |    |
|-------|-------|---|---|---|------|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0     | 1 | 2 | 3 | 4    | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | CHNUM |   |   |   | TYPE | 0 |   |   |   |   |    |    |    |    |    |    |
| W     |       |   |   |   |      |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0     | 0 | 0 | 0 | 0    | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |    |    |    |     |    |             |    |    |    |    |                 |    |    |    |    |
|-------|----|----|----|----|-----|----|-------------|----|----|----|----|-----------------|----|----|----|----|
| Bit   | 16 | 17 | 18 | 19 | 20  | 21 | 22          | 23 | 24 | 25 | 26 | 27              | 28 | 29 | 30 | 31 |
| R     | 0  |    |    |    | CFG |    | ID7_4_ID3_0 |    |    |    | 0  | ID3_0_DATA15_12 |    |    |    | 0  |
| W     |    |    |    |    |     |    |             |    |    |    |    |                 |    |    |    |    |
| Reset | 0  | 0  | 0  | 0  | 0   | 0  | 0           | 0  | 0  | 0  | 0  | 0               | 0  | 0  | 0  | 0  |

### SRX\_DMA\_SMSG\_BIT3 field descriptions

| Field                    | Description                                                                                                                                                            |
|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–3<br>CHNUM             | Channel Number. Indicates the channel number of the received message being read by DMA. Valid values are 0 to 5 .                                                      |
| 4<br>TYPE                | Serial Message Type. Indicates the type of Serial Message received<br>0 Short Serial Message<br>1 Enhanced Serial Message                                              |
| 5–20<br>Reserved         | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                |
| 21<br>CFG                | Configuration bit 'C'. Indicates the type of Enhanced Serial message<br>0 Enhanced Serial Message with 8-bit ID field<br>1 Enhanced Serial Message with 4-bit ID field |
| 22–25<br>ID7_4_ID3_0     | ID Field. Value depends on setting of 'C' bit. If 'C' bit is 0, this field is ID[7:4] and if the 'C' bit is 1, this field is ID[3:0]                                   |
| 26<br>Reserved           | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                |
| 27–30<br>ID3_0_DATA15_12 | ID or Data Field. Value depends on setting of 'C' bit. If 'C' bit is 0, this field is ID[3:0] and if the 'C' bit is 1, this field is Data[15:12]                       |
| 31<br>Reserved           | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                |

### 41.4.15 DMA Slow Serial Message Bit2 Read Register (SRX\_DMA\_SMSG\_BIT2)

This register (DMA\_SMSG\_BIT2) is common for all types of slow serial messages. Hence, the register fields have been placed in such a way to match the bit positions in actual message defined by SAE Specification as shown in [Table 41-83](#), [Table 41-84](#) and [Table 41-85](#). [Table 41-83](#), and [Table 41-84](#) shows the enhanced serial messages format to be used when reading this register for these two types of messages. [Table 41-85](#) shows the short serial message.

DMA Source Size must be set to 32 bits when reading this register via DMA.

Address: 0h base + 54h offset = 54h

|       |   |   |   |   |   |   |   |   |   |   |       |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|-------|----|----|----|----|----|----|----|----|----|------------|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10    | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20         | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   | SMCRC |    |    |    |    |    | 0  |    |    |    | DATA[11:0] |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |       |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0     | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**SRX\_DMA\_SMSG\_BIT2 field descriptions**

| Field               | Description                                                                                                                                                                                                                                                                     |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–9<br>Reserved     | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                         |
| 10–15<br>SMCRC      | 6-bit CRC value of the message. When TYPE bit in <a href="#">DMA Slow Serial Message Bit3 Read Register (SRX_DMA_SMSG_BIT3)</a> is set to 0, this CRC is of 4-bits for Short Serial Messages and when TYPE bit is set to 1, this CRC is of 6-bits for Enhanced Serial Messages. |
| 16–19<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                         |
| 20–31<br>DATA[11:0] | 12-bit Data Value                                                                                                                                                                                                                                                               |

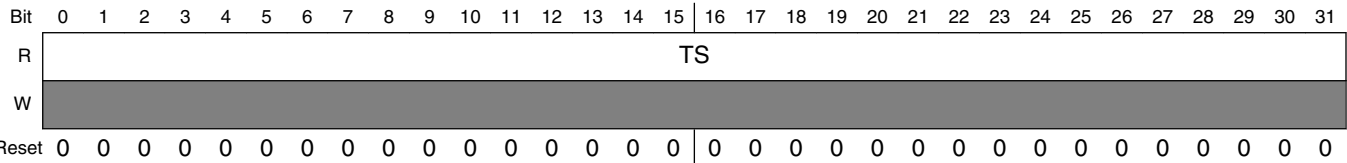
### 41.4.16 DMA Slow Serial Message Time Stamp Read Register (SRX\_DMA\_SMSG\_TS)

The resolution of the time stamp is set by programming the Time Stamp Prescaler Value in the [Global Control Register \(SRX\\_GBL\\_CTRL\)](#) register.

DMA Source Size must be set to 32 bits when reading this register (DMA\_SMSG\_TS) via DMA.

Memory map and register definition

Address: 0h base + 58h offset = 58h



SRX\_DMA\_SMSG\_TS field descriptions

| Field      | Description                                                                                                                                                                                                                                                                                                                                                                                                                         |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>TS | Time Stamp for the received message. Indicates the relative time of detection of the falling edge that comes at the end of the calibration pulse (which is also the start of the status and communication nibble pulse) of the last Fast Message that completed the Serial Message. Time Stamp for Slow Serial Messages is not latched when the Slow Serial Message starts but when the Slow Serial Message is completely received. |

41.4.17 Channel 'n' Clock Control Register (n = 0 to (CH-1)) (SRX\_CHn\_CLK\_CTRL)

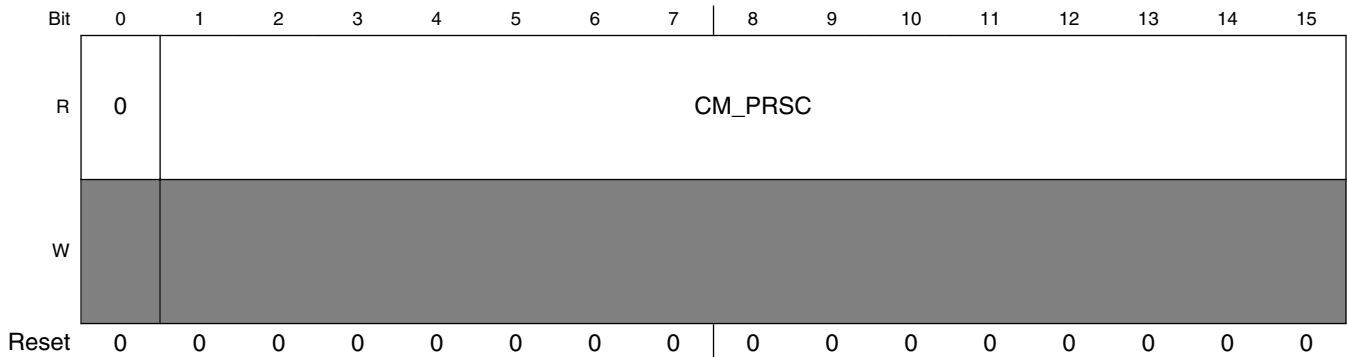
NOTE

The value of the CH is device-specific. See the device configuration section for details.

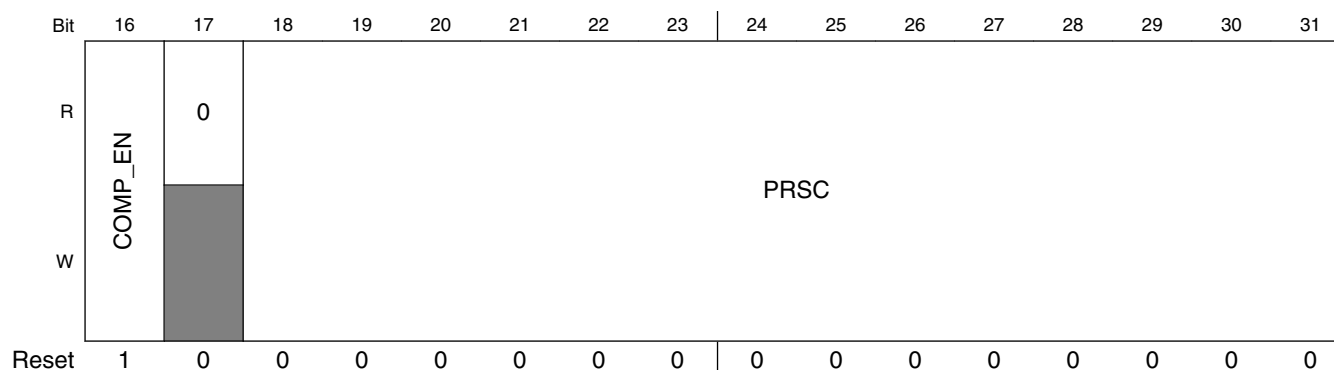
The SENT Receiver module generates the Receiver Sensor Clock (or simply Receiver Clock) internally to sample the incoming filtered sensor data. The user software must program the prescaler value (for Receiver Clock) to match the ideal Sensor Tx Clock period. The generated Receiver Clock is compensated for variations and clock drift that might occur in the Sensor Tx Clock. See [Adjustment for variation in sensor \(Tx\) clock](#) for more details.

This register configures the generation and subsequent compensation logic of the Receiver Clock. User must program the contents of this register before enabling the channel by writing to the corresponding channel enable bit in the [Channel Enable Register \(SRX\\_CHNL\\_EN\)](#) .

Address: 0h base + 60h offset + (16d × i), where i=0d to 5d







### SRX\_CHn\_CLK\_CTRL field descriptions

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 1–15<br>CM_PRSC | Compensated Prescaler Value. This is the compensated prescaler value calculated by Clock Compensation logic and this value is used to compensate for sensor Tx clock variation. This value is indicated as a read only field for use by user software, if needed. This is a 15-bit value.<br><br>'= Rx Prescaler Value' No variation, hence no compensation or Compensation is disabled<br>'< Rx Prescaler Value' Tx Clock is SLOWER than the programmed receive clock<br>'> Rx Prescaler Value' Tx Clock is FASTER than the programmed receive clock<br>Thus, the new Receiver Clock Tick Period can be computed by software as<br>Compensated Prescaler Value x High Frequency Receiver Clock Period<br>Note: The value provided in this field is ceil value of actual tick period as measured by the Receiver.                                                                                                                                                                                        |
| 16<br>COMP_EN   | Compensation Enable. To enable compensation logic to adjust the receiver clock against variation in Tx clock on this channel. User must set the Prescaler Value before enabling this bit.<br><br>0 Compensation is disabled<br>1 Compensation is enabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 17<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 18–31<br>PRSC   | Rx Prescaler Value. This factor is required to generate required Receiver Clock tick from the High Frequency Receiver Clock. This value can only be written when Channel Enable bit is Zero. Hence this value must be programmed before enabling the channel. If required receiver clock tick period (Trx_clk) is not divisible by the high frequency receiver clock (protocol clock) period (Thf_clk), the required prescaler value will have a fractional part which cannot be programmed. In this case then user is advised to program the prescaler value to the lower integer value.<br><br>0 Prescaler is bypassed. Should not be used.<br>Non-Zero Value Enables Prescaler<br>The required value is computed by software as follows:<br>$\text{Prescaler\_value} = \text{floor}\left(\frac{T_{\text{rx\_clk}}}{T_{\text{hf\_clk}}}\right)$ (both clock periods taken in μs).<br>Please see <a href="#">High frequency receiver clock (protocol clock) requirements</a> before setting this value. |

Table continues on the next page...

**SRX\_CHn\_CLK\_CTRL field descriptions (continued)**

| Field | Description                                                                                                                                                                                                                                       |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | e.g. if the required receiver clock tick period is 43 $\mu$ s and the High Frequency Receiver clock is selected to be 62.5 MHz (or 0.016 $\mu$ s) then,<br>Prescaler Value = $43/0.016 = 2687.5$ . Hence user must program 2687 in this register. |

#### 41.4.18 Channel 'n' Status Register (n=0 to (CH-1)) (SRX\_CHn\_STATUS)

This registers stores information about certain events and error conditions that occur in the channel during the message reception process. The assertion of these bit is not linked to any message but it is a general alert to the CPU that a particular condition (error or normal event) happened in the recent past. Messages are automatically discarded when an error is detected.

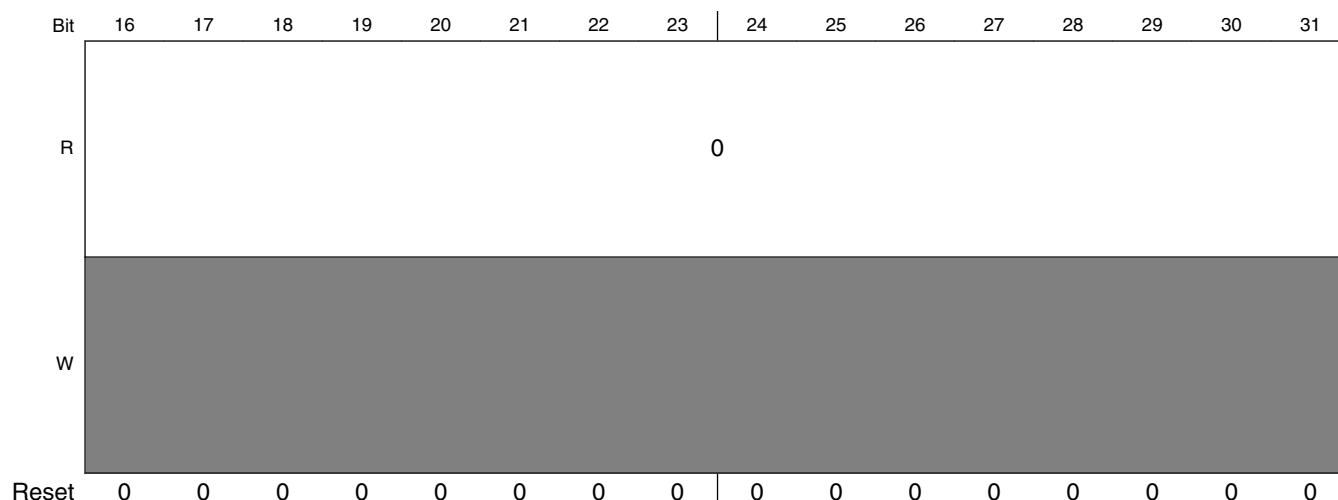
User software can clear these bit at any time by writing 1 to them. This register can be updated at any time irrespective of the setting in the [Channel Enable Register \(SRX\\_CHNL\\_EN\)](#) bit for the corresponding channel.

**NOTE**

The value of the CH is device-specific. See the device configuration section for details.

Address: 0h base + 64h offset + (16d  $\times$  i), where i=0d to 5d

| Bit   | 0        | 1 | 2 | 3 | 4          | 5         | 6         | 7         | 8 | 9           | 10          | 11           | 12          | 13           | 14           | 15            |
|-------|----------|---|---|---|------------|-----------|-----------|-----------|---|-------------|-------------|--------------|-------------|--------------|--------------|---------------|
| R     | BUS_IDLE | 0 |   |   | CAL_RESYNC | CAL_20_25 | SMSG_OFLW | FMSG_OFLW | 0 | PP_DIAG_ERR | CAL_LEN_ERR | CAL_DIAG_ERR | NIB_VAL_ERR | SMSG_CRC_ERR | FMSG_CRC_ERR | NUM_EDGES_ERR |
| W     | w1c      |   |   |   | w1c        | w1c       | w1c       | w1c       |   | w1c         | w1c         | w1c          | w1c         | w1c          | w1c          | w1c           |
| Reset | 0        | 0 | 0 | 0 | 0          | 0         | 0         | 0         | 0 | 0           | 0           | 0            | 0           | 0            | 0            | 0             |



### SRX\_CHn\_STATUS field descriptions

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>BUS_IDLE   | <p>Bus Idle Status. This bit indicates that the sensor interface has been idle for more than the period defined by the programmed Bus Idle Count value(in <a href="#">Channel 'n' Configuration Register (n=0 to (CH-1)) (SRX_CHn_CONFIG)</a> ). CPU should write 1 to this clear this bit once it is read.</p> <p>0 Bus is not idle<br/>1 Channel has been idle for more than the allowed value</p>                                                                                             |
| 1-3<br>Reserved | <p>This field is reserved.<br/>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                  |
| 4<br>CAL_RESYNC | <p>Successive Calibration Check (Option 2 of SAE SENT Spec) Resynchronized. This bit indicates that Successive Calibration Check (Option 2 of SAE Spec) has failed three times and has resynchronized to assume 3rd message to be correct and will be received. The successive calibration check error will also be asserted during re-synchronization (to indicate the 3rd error being detected).</p> <p>0 No interrupt<br/>1 Interrupt Status condition has occurred</p>                       |
| 5<br>CAL_20_25  | <p>Calibration Variation 20 - 25% Interrupt Status. This bit indicates that the calibration pulse received on this channel has variation in between <math>\pm 20\%</math> to <math>\pm 25\%</math> from 56 ticks.</p> <p>0 No Interrupt<br/>1 Interrupt Status condition has occurred</p>                                                                                                                                                                                                        |
| 6<br>MSG_OFLW   | <p>Slow Serial Message Overflow Status. This bit indicates overflow occurred on this channel. Overflow will cause state machine to halt stopping reception of new messages.</p> <p>0 No Interrupt<br/>1 Interrupt Status condition has occurred</p>                                                                                                                                                                                                                                              |
| 7<br>FMSG_OFLW  | <p>Fast Message Overflow Status. This bit indicates overflow occurred on this channel. Overflow happens when effective data read rate by CPU is slower than data receive rate on channel. On overflow, new incoming messages will overwrite the previous messages (that are not read out) stored in the receiver buffers. However, any message that is currently being read out by DMA or CPU will not get over-written.</p> <p>0 No Interrupt<br/>1 Interrupt Status condition has occurred</p> |

Table continues on the next page...

**SRX\_CHn\_STATUS field descriptions (continued)**

| Field               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 8<br>Reserved       | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                    |
| 9<br>PP_DIAG_ERR    | This diagnostic checks status bit indicates that the ratio of calibration pulse length to overall message length (with pause pulse) is more than $\pm 1.5625\%$ between two messages. This check is valid only for messages with pause pulse. This bit indicates whether this diagnostic has failed or passed on this channel, and is cleared by writing a 1 to it.<br><br>0 Error Check has passed<br>1 Error Check has failed                                                            |
| 10<br>CAL_LEN_ERR   | This diagnostic checks status bit indicates that Calibration pulse is more than 56 ticks $\pm 25\%$ . This bit indicates whether this diagnostic has failed or passed on this channel, and is cleared by writing a 1 to it.<br><br>0 Error Check has passed<br>1 Error Check has failed                                                                                                                                                                                                    |
| 11<br>CAL_DIAG_ERR  | Successive Calibration pulses differ by $\pm 1.56\%$ . SAE spec defines it be 1.5625% but the accuracy of this check depends on the frequency of protocol clock (See <a href="#">High frequency receiver clock (protocol clock) requirements</a> ). This diagnostic checks status bit indicates whether this diagnostic has failed or passed on this channel, and is cleared by writing a 1 to it.<br><br>0 Error Check has passed<br>1 Error Check has failed                             |
| 12<br>NIB_VAL_ERR   | Any nibble data value $<0$ or $>15$ . This diagnostic checks status bit indicates whether this diagnostic has failed or passed on this channel, and is cleared by writing a 1 to it.<br><br>0 Error Check has passed<br>1 Error Check has failed                                                                                                                                                                                                                                           |
| 13<br>MSG_CRC_ERR   | Checksum error in Slow Serial Message. This diagnostic checks status bit indicates whether this diagnostic has failed or passed on this channel, and is cleared by writing a 1 to it.<br><br>0 Error Check has passed<br>1 Error Check has failed                                                                                                                                                                                                                                          |
| 14<br>FMSG_CRC_ERR  | Checksum error in Fast Message. This diagnostic checks status bit indicates whether this diagnostic has failed or passed on this channel, and is cleared by writing a 1 to it.<br><br>0 Error Check has passed<br>1 Error Check has failed                                                                                                                                                                                                                                                 |
| 15<br>NUM_EDGES_ERR | Not the expected number of negative edges between calibration pulse. This diagnostic checks status bit indicates whether this diagnostic has failed or passed on this channel. This bit will not set in case of "Option 2" for successive calibration pulse check is selected. This error will only set when "Option 1" for successive calibration check is selected (SUCC_CAL_CHK). This bit is cleared by writing a 1 to it.<br><br>0 Error Check has passed<br>1 Error Check has failed |
| 16–31<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                    |

### 41.4.19 Channel 'n' Configuration Register (n=0 to (CH-1)) (SRX\_CHn\_CONFIG)

This register (CHn\_CONFIG) provides the configurability for interrupt assertion for various events and error as explained in [Channel 'n' Status Register \(n=0 to \(CH-1\)\) \(SRX\\_CHn\\_STATUS\)](#) and control bits for selection of various receive options that might be required by the user software. This register also configures the width of noise glitches (on the sensor input) to be filtered out by the input programmable filter.

User must program the contents of this register before enabling the channel by writing to the corresponding channel enable bit in the [Channel Enable Register \(SRX\\_CHNL\\_EN\)](#).

#### NOTE

The value of the CH is device-specific. See the device configuration section for details.

Address: 0h base + 68h offset + (16d × i), where i=0d to 5d

| Bit   | 0            | 1       | 2         | 3         | 4             | 5            | 6            | 7            | 8            | 9              | 10             | 11              | 12             | 13              | 14              | 15               |
|-------|--------------|---------|-----------|-----------|---------------|--------------|--------------|--------------|--------------|----------------|----------------|-----------------|----------------|-----------------|-----------------|------------------|
| R     | BUS_IDLE_CNT |         |           |           | IE_CAL_RESYNC | IE_CAL_20_25 | IE_SMSG_OFLW | IE_FMSG_OFLW | FCRC_CHK_OFF | IE_PP_DIAG_ERR | IE_CAL_LEN_ERR | IE_CAL_DIAG_ERR | IE_NIB_VAL_ERR | IE_SMSG_CRC_ERR | IE_FMSG_CRC_ERR | IE_NUM_EDGES_ERR |
| W     |              |         |           |           |               |              |              |              |              |                |                |                 |                |                 |                 |                  |
| Reset | 0            | 0       | 0         | 0         | 0             | 0            | 0            | 0            | 0            | 0              | 0              | 0               | 0              | 0               | 0               | 0                |
| Bit   | 16           | 17      | 18        | 19        | 20            | 21           | 22           | 23           | 24           | 25             | 26             | 27              | 28             | 29              | 30              | 31               |
| R     | DCHNG_INT    | CAL_RNG | PP_CHKSEL | FCRC_TYPE | FCRC_SC_EN    | SCRC_TYPE    | PAUSE_EN     | SUCC_CAL_CHK | FIL_CNT      |                |                |                 |                |                 |                 |                  |
| W     |              |         |           |           |               |              |              |              |              |                |                |                 |                |                 |                 |                  |
| Reset | 0            | 0       | 0         | 0         | 0             | 0            | 0            | 1            | 0            | 0              | 0              | 0               | 0              | 1               | 0               | 0                |

#### SRX\_CHn\_CONFIG field descriptions

| Field               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |      |           |      |                                  |      |                                  |      |                                  |      |                                   |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|-----------|------|----------------------------------|------|----------------------------------|------|----------------------------------|------|-----------------------------------|
| 0–3<br>BUS_IDLE_CNT | <p>Bus Idle Count. This register value defines the maximum allowable idle period on the sensor interface of this channel. The value is defined as follows:</p> <table> <tr> <td>0000</td><td>Disabled.</td></tr> <tr> <td>0001</td><td>127*2 Receiver Clock Tick Counts</td></tr> <tr> <td>0010</td><td>127*4 Receiver Clock Tick Counts</td></tr> <tr> <td>0100</td><td>127*8 Receiver Clock Tick Counts</td></tr> <tr> <td>1000</td><td>127*16 Receiver Clock Tick Counts</td></tr> </table> | 0000 | Disabled. | 0001 | 127*2 Receiver Clock Tick Counts | 0010 | 127*4 Receiver Clock Tick Counts | 0100 | 127*8 Receiver Clock Tick Counts | 1000 | 127*16 Receiver Clock Tick Counts |
| 0000                | Disabled.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |      |           |      |                                  |      |                                  |      |                                  |      |                                   |
| 0001                | 127*2 Receiver Clock Tick Counts                                                                                                                                                                                                                                                                                                                                                                                                                                                               |      |           |      |                                  |      |                                  |      |                                  |      |                                   |
| 0010                | 127*4 Receiver Clock Tick Counts                                                                                                                                                                                                                                                                                                                                                                                                                                                               |      |           |      |                                  |      |                                  |      |                                  |      |                                   |
| 0100                | 127*8 Receiver Clock Tick Counts                                                                                                                                                                                                                                                                                                                                                                                                                                                               |      |           |      |                                  |      |                                  |      |                                  |      |                                   |
| 1000                | 127*16 Receiver Clock Tick Counts                                                                                                                                                                                                                                                                                                                                                                                                                                                              |      |           |      |                                  |      |                                  |      |                                  |      |                                   |

Table continues on the next page...

**SRX\_CHn\_CONFIG field descriptions (continued)**

| Field                     | Description                                                                                                                                                                                                                                                                                                                                                                                                          |
|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 4<br>IE_CAL_<br>RESYNC    | Successive Calibration Check Resynchronized Interrupt Enable: This bit enables interrupt assertion when Successive Calibration diagnosis has failed three times in case of "Option 2" being selected for Successive Calibration Check Method and the check resynchronizes the take the third message to be correct.<br><br>0 Interrupt is disabled<br>1 Interrupt is enabled                                         |
| 5<br>IE_CAL_20_25         | Calibration Variation 20 - 25% Interrupt Enable. This bit enables interrupt assertion when corresponding status bit is set.<br><br>0 Interrupt is disabled<br>1 Interrupt is enabled                                                                                                                                                                                                                                 |
| 6<br>IE_SMSG_OFLW         | Slow Serial Message Overflow Interrupt Enable. This bit enables interrupt assertion when overflow occurs on reception of Slow Serial Messages in corresponding channel<br><br>0 Interrupt is disabled<br>1 Interrupt is enabled                                                                                                                                                                                      |
| 7<br>IE_FMSG_OFLW         | Fast Message Overflow Interrupt Enable. This bit enables interrupt assertion when overflow occurs on reception of Fast Messages in corresponding channel.<br><br>0 Interrupt is disabled<br>1 Interrupt is enabled                                                                                                                                                                                                   |
| 8<br>FCRC_CHK_<br>OFF     | Fast Message CRC Check Off: This bit can be used to switch off CRC check in Fast Message<br><br>0 Check is enabled<br>1 Check is disabled/off                                                                                                                                                                                                                                                                        |
| 9<br>IE_PP_DIAG_<br>ERR   | Ratio of calibration pulse length to message length varies by more than $\pm 1.5625\%$ between two frames. Valid for messages with pause pulse. This bit enables interrupt assertion when its respective diagnostic check fails on the corresponding channel.<br><br>0 Interrupt is disabled<br>1 Interrupt is enabled                                                                                               |
| 10<br>IE_CAL_LEN_<br>ERR  | Calibration pulse is wider than 56 ticks $\pm 25\%$ . This bit enables interrupt assertion when its respective diagnostic check fails on the corresponding channel.<br><br>0 Interrupt is disabled<br>1 Interrupt is enabled                                                                                                                                                                                         |
| 11<br>IE_CAL_DIAG_<br>ERR | Successive Calibration pulses differ by more than $\pm 1.56\%$ . SAE spec defines it be 1.5625% but the accuracy of this check depends on the protocol clock. For instance for 62.5 MHz, this check will fail for 1.56% and passes for 1.55%. This bit enables interrupt assertion when its respective diagnostic check fails on the corresponding channel.<br><br>0 Interrupt is disabled<br>1 Interrupt is enabled |
| 12<br>IE_NIB_VAL_<br>ERR  | Any nibble data value $<0$ or $>15$ . This bit enables interrupt assertion when its respective diagnostic check fails on the corresponding channel.<br><br>0 Interrupt is disabled<br>1 Interrupt is enabled                                                                                                                                                                                                         |

*Table continues on the next page...*

**SRX\_CHn\_CONFIG field descriptions (continued)**

| Field                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 13<br>IE_SMSG_CRC_ERR  | Checksum error in Slow Serial Message. This bit enables interrupt assertion when its respective diagnostic check fails on the corresponding channel.<br><br>0 Interrupt is disabled<br>1 Interrupt is enabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 14<br>IE_FMSG_CRC_ERR  | Checksum error in Fast Message. This bit enables interrupt assertion when its respective diagnostic check fails on the corresponding channel.<br><br>0 Interrupt is disabled<br>1 Interrupt is enabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 15<br>IE_NUM_EDGES_ERR | Not the expected number of negative edges between calibration pulse. This bit enables interrupt assertion when its respective diagnostic check fails on the corresponding channel.<br><br>0 Interrupt is disabled<br>1 Interrupt is enabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 16<br>DCHNG_INT        | Enable for Interrupt on Reception of Fast Message with Changed Data: This bit is used to enable storing and generating corresponding DMA or Message Ready Interrupt Request for Fast Message only when any of the data nibbles after different from previously received Fast Message on that channel. If a new Fast Message is received that has same values for all data nibbles as that of previous message then the current message will be discarded and will not be stored in the buffer, or FIFO, and neither Fast Message Ready Interrupt nor DMA request will be generated.<br><br>0 All Fast Messages will be received even if data nibbles are same<br>1 Only Fast Messages with differing values of data nibbles will be received |
| 17<br>CAL_RNG          | Valid Calibration Pulse Range Selection: This bit is used to control whether 20% variation or 25% variation in Calibration Pulse will be tolerated when detecting a Calibration Pulse<br><br>0 20% variation is acceptable<br>1 25% variation is acceptable                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 18<br>PP_CHKSEL        | Pause Pulse Diagnostic Check Selection: This bit controls which diagnostic checks to run when messages are enabled to be received with pause pulses.<br><br>0 Both Successive Calibration Pulse Check and Pause Pulse Diagnostic are run<br>1 Only Pause Pulse Diagnostic is run. Successive Calibration Pulse Check is disabled.                                                                                                                                                                                                                                                                                                                                                                                                            |
| 19<br>FCRC_TYPE        | Fast Message CRC Type. This indicates the type of CRC method to be used for CRC Diagnostic Check on the received fast messages<br><br>0 Recommended implementation (additional 0 data nibble XORed with the rest of the nibbles)<br>1 Legacy implementation (no additional 0 data nibble is XORed)                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 20<br>FCRC_SC_EN       | Fast Message CRC Status and Communication Nibble Enable. This bit enables the inclusion of Status and Communication Nibble when CRC is calculated for Fast Messages.<br><br>0 Status and Communication Nibble not included in CRC calculation<br>1 Status and Communication Nibble is included in CRC calculation                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 21<br>SCRC_TYPE        | Slow Serial Message CRC Type. This indicates the type of CRC method to be used for CRC Diagnostic Check on the received slow serial messages.<br><br>0 Recommended implementation (additional 0 data nibble XORed with the rest of the nibbles)<br>1 Legacy implementation (no additional 0 data nibble XORed)                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 22<br>PAUSE_EN         | Pause Pulse Enable. Enables the channel receiver to detect a pause pulse.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

*Table continues on the next page...*

**SRX\_CHn\_CONFIG field descriptions (continued)**

| Field              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                    | 0 Detection of Pause Pulse is disabled<br>1 Detection of Pause Pulse enabled                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 23<br>SUCC_CAL_CHK | Successive Calibration Pulse Check Method. This bit indicates the method to be used for performing the successive calibration pulse check. Default value is 1.<br><br>0 Option 2 i.e. Low Latency Option as per SAE Specification<br>1 Option 1 i.e. Preferred but High Latency Option as per SAE Specification                                                                                                                                                                                                                                       |
| 24–31<br>FIL_CNT   | Input Filter Sample Count. This indicates twice the number of Protocol Clock (of High Frequency Receiver Clock) cycles required for channel input from device's pad to remain stable before it is sampled as 0 or 1. This field defines the width of glitches to be filtered out by the input programmable filter on that channel. Only one bit should be set in this field. Output of the filter is equal to $2 \times \text{FIL\_CNT} + 4$ .<br><br>0 No filtering. Input from device's pad is only synchronized<br>1 Non-Zero Filtering is enabled |

**41.4.20 Channel 'n' Fast Message Data Read Register (n=0 to (CH-1)) (SRX\_CHn\_FMSG\_DATA)**

Address offset:  $0x0160 + 0x18 \times n$

This is register CHn\_FMSG\_DATA. The contents of this register are same as [DMA Fast Message Data Read Register \(SRX\\_DMA\\_FMSG\\_DATA\)](#).

**NOTE**

The value of the CH is device-specific. See the device configuration section for details.

Address:  $0h \text{ base} + 160h \text{ offset} + (24d \times i)$ , where  $i=0d$  to  $5d$

|       |       |   |   |   |       |   |   |   |       |   |    |    |       |    |    |    |       |    |    |    |       |    |    |    |       |    |    |    |       |    |    |    |
|-------|-------|---|---|---|-------|---|---|---|-------|---|----|----|-------|----|----|----|-------|----|----|----|-------|----|----|----|-------|----|----|----|-------|----|----|----|
| Bit   | 0     | 1 | 2 | 3 | 4     | 5 | 6 | 7 | 8     | 9 | 10 | 11 | 12    | 13 | 14 | 15 | 16    | 17 | 18 | 19 | 20    | 21 | 22 | 23 | 24    | 25 | 26 | 27 | 28    | 29 | 30 | 31 |
| R     | CHNUM |   |   |   | SCNIB |   |   |   | DNIB1 |   |    |    | DNIB2 |    |    |    | DNIB3 |    |    |    | DNIB4 |    |    |    | DNIB5 |    |    |    | DNIB6 |    |    |    |
| W     |       |   |   |   |       |   |   |   |       |   |    |    |       |    |    |    |       |    |    |    |       |    |    |    |       |    |    |    |       |    |    |    |
| Reset | 0     | 0 | 0 | 0 | 0     | 0 | 0 | 0 | 0     | 0 | 0  | 0  | 0     | 0  | 0  | 0  | 0     | 0  | 0  | 0  | 0     | 0  | 0  | 0  | 0     | 0  | 0  | 0  | 0     | 0  | 0  | 0  |

**SRX\_CHn\_FMSG\_DATA field descriptions**

| Field          | Description                                                                                                      |
|----------------|------------------------------------------------------------------------------------------------------------------|
| 0–3<br>CHNUM   | Channel Number. Indicates the channel number of the received message being read by DMA. Valid values are 0 to 5. |
| 4–7<br>SCNIB   | Status and Communication Nibble of message                                                                       |
| 8–11<br>DNIB1  | Data Nibble 1. Always supported                                                                                  |
| 12–15<br>DNIB2 | Data Nibble 2. Should be ignored if number of data nibbles supported by channel is 1                             |

Table continues on the next page...



**SRX\_CHn\_FMSG\_DATA field descriptions (continued)**

| Field          | Description                                                                                  |
|----------------|----------------------------------------------------------------------------------------------|
| 16–19<br>DNIB3 | Data Nibble 3. Should be ignored if number of data nibbles supported by channel is 2 or less |
| 20–23<br>DNIB4 | Data Nibble 4. Should be ignored if number of data nibbles supported by channel is 3 or less |
| 24–27<br>DNIB5 | Data Nibble 5. Should be ignored if number of data nibbles supported by channel is 4 or less |
| 28–31<br>DNIB6 | Data Nibble 6. Should be ignored if number of data nibbles supported by channel is 5 or less |

**41.4.21 Channel 'n' Fast Message CRC Read Register (n=0 to (CH-1)) (SRX\_CHn\_FMSG\_CRC)**

Address offset:  $0x0164 + 0x18 \times n$

This is register CHn\_FMSG\_CRC. The contents of this register are same as [DMA Fast Message CRC Read Register \(SRX\\_DMA\\_FMSG\\_CRC\)](#).

Address:  $0h \text{ base} + 164h \text{ offset} + (24d \times i)$ , where  $i=0d$  to  $5d$

|       |   |   |   |   |   |   |   |   |   |   |    |    |       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12    | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    | CRC4b |    |    |    | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0     | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**SRX\_CHn\_FMSG\_CRC field descriptions**

| Field             | Description                                                                             |
|-------------------|-----------------------------------------------------------------------------------------|
| 0–11<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 12–15<br>CRC4b    | 4-bit CRC value of the message.                                                         |
| 16–31<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |

**41.4.22 Channel 'n' Fast Message Time Stamp Read Register (n=0 to (CH-1)) (SRX\_CHn\_FMSG\_TS)**

Address offset:  $0x0168 + 0x18 \times n$

This is register CHn\_FMSG\_TS. The contents of this register are same as [DMA Fast Message Time Stamp Read Register \(SRX\\_DMA\\_FMSG\\_TS\)](#).

## Memory map and register definition

Address: 0h base + 168h offset + (24d × i), where i=0d to 5d

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    | TS |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

### SRX\_CHn\_FMSG\_TS field descriptions

| Field      | Description                                                                                                                                                                                                           |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>TS | Time Stamp for received message. Indicates the relative time when the falling edge that comes at the end of the calibration pulse (which is also the start of the status and communication nibble pulse) was detected |

## 41.4.23 Channel 'n' Serial Message Read Register (Bit 3) (n=0 to (CH-1)) (SRX\_CHn\_SMSG\_BIT3)

Address offset: 0x016C + 0x18 × n

This is register CHn\_SMSG\_BIT3. The contents of this register are same as [DMA Slow Serial Message Bit3 Read Register \(SRX\\_DMA\\_SMSG\\_BIT3\)](#) .

### NOTE

The value of the CH is device-specific. See the device configuration section for details.

Address: 0h base + 16Ch offset + (24d × i), where i=0d to 5d

|       |       |   |   |   |      |   |   |   |   |   |    |    |    |    |    |    |
|-------|-------|---|---|---|------|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0     | 1 | 2 | 3 | 4    | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | CHNUM |   |   |   | TYPE | 0 |   |   |   |   |    |    |    |    |    |    |
| W     |       |   |   |   |      |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0     | 0 | 0 | 0 | 0    | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |    |    |    |     |             |    |    |    |    |                 |    |    |    |    |    |
|-------|----|----|----|----|-----|-------------|----|----|----|----|-----------------|----|----|----|----|----|
| Bit   | 16 | 17 | 18 | 19 | 20  | 21          | 22 | 23 | 24 | 25 | 26              | 27 | 28 | 29 | 30 | 31 |
| R     | 0  |    |    |    | CFG | ID7_4_ID3_0 |    |    |    | 0  | ID3_0_DATA15_12 |    |    |    | 0  |    |
| W     |    |    |    |    |     |             |    |    |    |    |                 |    |    |    |    |    |
| Reset | 0  | 0  | 0  | 0  | 0   | 0           | 0  | 0  | 0  | 0  | 0               | 0  | 0  | 0  | 0  | 0  |

### SRX\_CHn\_SMSG\_BIT3 field descriptions

| Field        | Description                                                                                                               |
|--------------|---------------------------------------------------------------------------------------------------------------------------|
| 0–3<br>CHNUM | Channel Number. Indicates the channel number of the received message being read by DMA. Valid values are 0 to 5 .         |
| 4<br>TYPE    | Serial Message Type. Indicates the type of Serial Message received<br>0 Short Serial Message<br>1 Enhanced Serial Message |

Table continues on the next page...

**SRX\_CHn\_SMSG\_BIT3 field descriptions (continued)**

| Field                    | Description                                                                                                                                                            |
|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 5–20<br>Reserved         | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                |
| 21<br>CFG                | Configuration bit 'C'. Indicates the type of Enhanced Serial message<br>0 Enhanced Serial Message with 8-bit ID field<br>1 Enhanced Serial Message with 4-bit ID field |
| 22–25<br>ID7_4_ID3_0     | ID Field. Value depends on setting of 'C' bit. If 'C' bit is 0, this field is ID[7:4] and if the 'C' bit is 1, this field is ID[3:0]                                   |
| 26<br>Reserved           | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                |
| 27–30<br>ID3_0_DATA15_12 | ID or Data Field. Value depends on setting of 'C' bit. If 'C' bit is 0, this field is ID[3:0] and if the 'C' bit is 1, this field is Data[15:12]                       |
| 31<br>Reserved           | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                |

**41.4.24 Channel 'n' Serial Message Read Register (Bit 2)(n=0 to (CH-1)) (SRX\_CHn\_SMSG\_BIT2)**

Address offset:  $0x0170 + 0x18 \times n$

This is register CHn\_SMSG\_BIT2. The contents of this register are same as [DMA Slow Serial Message Bit2 Read Register \(SRX\\_DMA\\_SMSG\\_BIT2\)](#) .

Address:  $0h \text{ base} + 170h \text{ offset} + (24d \times i)$ , where  $i=0d$  to  $5d$

|       |   |   |   |   |   |   |   |   |   |   |       |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|-------|----|----|----|----|----|----|----|----|----|------------|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10    | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20         | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   | SMCRC |    |    |    |    |    | 0  |    |    |    | DATA[11:0] |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |       |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0     | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**SRX\_CHn\_SMSG\_BIT2 field descriptions**

| Field               | Description                                                                                                                                                                                                                                                                     |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–9<br>Reserved     | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                         |
| 10–15<br>SMCRC      | 6-bit CRC value of the message. When TYPE bit in <a href="#">DMA Slow Serial Message Bit3 Read Register (SRX_DMA_SMSG_BIT3)</a> is set to 0, this CRC is of 4-bits for Short Serial Messages and when TYPE bit is set to 1, this CRC is of 6-bits for Enhanced Serial Messages. |
| 16–19<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                         |
| 20–31<br>DATA[11:0] | 12-bit Data Value                                                                                                                                                                                                                                                               |

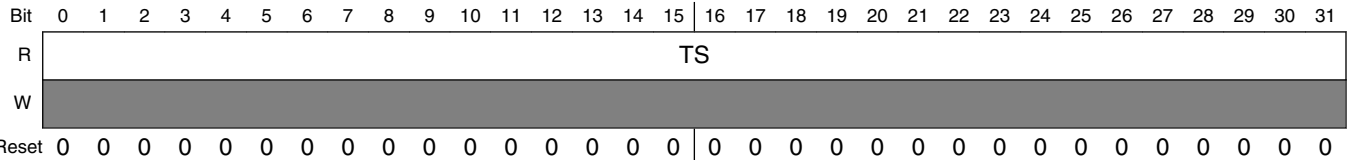
41.4.25

Channel 'n' Serial Message Time Stamp Read Register (n=0 to (CH-1)) (SRX\_CHn\_SMSG\_TS)

Address offset: 0x0174 + 0x18 x n

This is register CHn\_SMSG\_TS. The contents of this register are same as [DMA Slow Serial Message Time Stamp Read Register \(SRX\\_DMA\\_SMSG\\_TS\)](#).

Address: 0h base + 174h offset + (24d × i), where i=0d to 5d



SRX\_CHn\_SMSG\_TS field descriptions

| Field      | Description                                                                                                                                                                                                                                                                                                                                                                                                                         |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>TS | Time Stamp for the received message. Indicates the relative time of detection of the falling edge that comes at the end of the calibration pulse (which is also the start of the status and communication nibble pulse) of the last Fast Message that completed the Serial Message. Time Stamp for Slow Serial Messages is not latched when the Slow Serial Message starts but when the Slow Serial Message is completely received. |

41.5

Functional description

This section describes the function of the SRX.

41.5.1

Initialization sequence

All the channels are required to be configured before enabling them by writing into corresponding channel enable bit in the [Channel Enable Register \(SRX\\_CHNL\\_EN\)](#). There are some registers which must be configured and some which are to be changed if reset value is not as required for that channel. After enabling the channel, no configuration should be changed and if required to be changed, the user needs to first disable the channel and then reconfigure the channel parameters. The following must to be configured before enabling the corresponding channel:

- Time Stamp Prescaler Value in [Global Control Register \(SRX\\_GBL\\_CTRL\)](#) to generate the time stamping clock

- Rx Prescaler Value in [Channel 'n' Clock Control Register \(n = 0 to \(CH-1\)\) \(SRX\\_CHn\\_CLK\\_CTRL\)](#) to generate the channel's receiver clock
- Compensation enable in [Channel 'n' Clock Control Register \(n = 0 to \(CH-1\)\) \(SRX\\_CHn\\_CLK\\_CTRL\)](#)
- Channel Configuration settings in the [Channel 'n' Configuration Register \(n=0 to \(CH-1\)\) \(SRX\\_CHn\\_CONFIG\)](#)
- Number of Data Nibbles in Channel in the [Data Control Register 1 \(SRX\\_DATA\\_CTRL1\)](#)
- Control bits that control whether messages received in channel are to read by interrupt or DMA in [Fast Message DMA Control Register \(SRX\\_FDMA\\_CTRL\)](#), [Slow Serial Message DMA Control Register \(SRX\\_SDMA\\_CTRL\)](#), [Fast Message Ready Interrupt Control Register \(SRX\\_FRDY\\_IE\)](#) and [Slow Serial Message DMA Control Register \(SRX\\_SDMA\\_CTRL\)](#)

## 41.5.2 DMA read logic

The SENT Receiver module generates separate DMA requests for all Fast Messages and Slow Serial Messages received. The DMA requests are common for all channels. Each packet stored in the channels' buffers are appended with a time stamp taken from a free running time keeping counter and the channel number on which it was received. Details on this time stamp counter can be found in [Time stamp logic](#).

### 41.5.2.1 DMA request for Fast Message reading via FIFO

A single DMA request is generated for all Fast Messages received on any channel. There is a DMA read FIFO to store the incoming messages from all the channels and allow them to be read via DMA. The DMA operation will happen as follows:

- DMA request will be asserted when number of messages stored in FIFO crosses the configured watermark level (in [Global Control Register \(GBL\\_CTRL\)](#)).
- DMA request will be de-asserted when number of messages stored goes below watermark level. It is assumed that DMA Controller's TCD would be set to transfer bytes equal to the messages indicated by watermark level so as to empty the FIFO.
- It is assumed that when responding to a DMA request, the DMA controller will read at a rate faster than the rate at which messages are getting received and stored in FIFO.

- The complete received message can be read by three consecutive read accesses to DMA Fast Message Data Read Register. The message will be read out in following sequence:
  - Fast Message data as described by DMA Fast Message Data Read Register (DMA\_FMSG\_DATA)
  - CRC for the Fast Message as described by DMA Fast Message CRC Read Register (DMA\_FMSG\_CRC)
  - Time stamp value for the Fast Message as described by DMA Fast Message Time Stamp Read Register (DMA\_FMSG\_TS)
- The channel number is prefixed in each valid message (no error) to allow software to identify messages pertaining to a channel
- [Table 41-82](#) shows the format in which the messages will be stored in any memory-mapped message buffer and when being read out from FIFO.
- Since multiple messages can be received at the same time, there is a [Fixed round robin sequencing logic](#), which will scan all active channels configured for DMA and store their message into the FIFO one by one. Channel configured for DMA but do not have ready messages will be automatically skipped till their turn comes again in the round robin sequence.
- Messages received can be stored in a different order to which they are actually received. The time stamp will indicate the order of reception of messages in a particular channel. However, the time stamp cannot be used to determine the order across channels.

This figure below shows the data for the three 32-bit accesses to be made by DMA to get one complete message with time stamp.

**Table 41-82. Format of storing of Fast Messages in FIFO**

| CH NUM<br>[3:0]   | Status and<br>Comm [3:0] | Data 1 [3:0] | Data 2 [3:0] | Data 3 [3:0] | Data 4 [3:0] | Data 5 [3:0] | Data 6 [3:0] |
|-------------------|--------------------------|--------------|--------------|--------------|--------------|--------------|--------------|
|                   |                          |              | CRC [3:0]    |              |              |              |              |
| TIME STAMP [31:0] |                          |              |              |              |              |              |              |

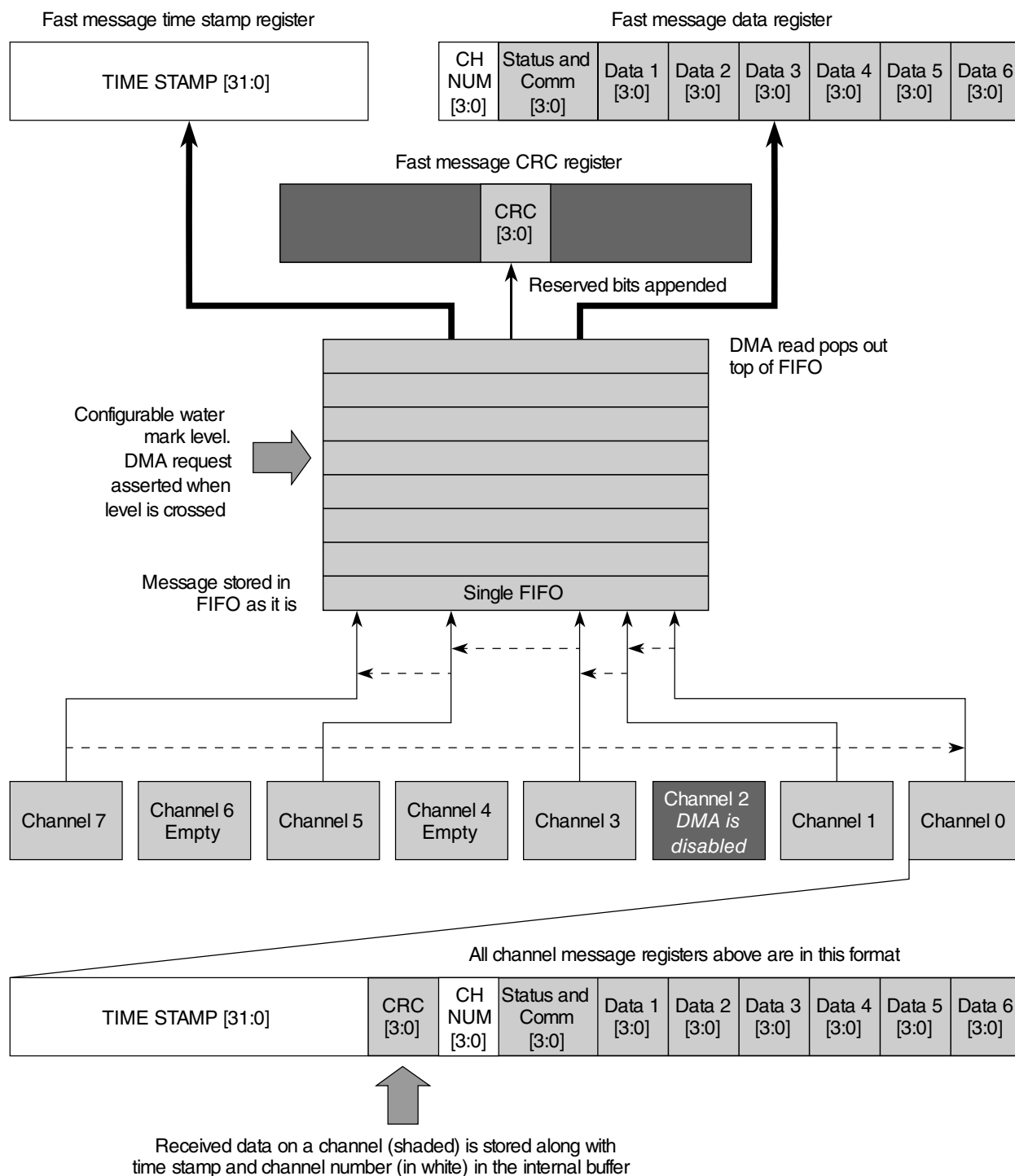


Figure 41-81. Fast Message DMA Read Logic using FIFO

### 41.5.2.1.1 Fixed round robin sequencing logic

Figure 41-81 shows a scenario where 8 channels (0-7) with all channels are enabled for DMA reading except channel 2. DMA request will be asserted when FIFO has received packets more than watermark level set. Figure 41-81 shows channels 6 and 4 have not received the complete message. The following points describe the complete round robin sequencing scheme.

- The round robin sequencing logic which stores the messages in the FIFO by looping through each channels' Fast Message register and checking for the availability of the message.
- As shown in Figure 41-81, the arbitration logic will check for messages in sequence 0 -> 1 -> 3 -> 5 -> 7 and back to channel 0 and keep looping. Channel 2 is skipped as it's not enabled to be read via DMA) and channel 4 and 6 are skipped as message is not received completely at the instance the sequencing logic checked their buffers. This sequence continues in this fashion.
- The sequence in which messages are stored in FIFO is not deterministic and depends on the sequence they are detected after the noise filtering by the input filter of the channel.
- The DMA controller responds to the DMA request by reading from the DMA Fast Message Read register. When a DMA read happens, a message is read from the FIFO and split into three 32-bit values and provided to DMA in the order as described in previous section. The three 32-bit values together form one complete Fast Message.
- Once all three 32-bit access are completed, the next data from the FIFO is popped.

Figure 41-81 also shows how the received data (light gray colored boxes) from a channel is stored in FIFO along with the time stamp and channel number appended (both white colored).

### 41.5.2.2 DMA request for Fast Message reading when FIFO is disabled

In this mode no FIFO will be used as it is disabled via the FIFO Enable bit in the Global Control Register (GBL\_CTRL). Fast Messages received are stored in a single system bus clock domain buffer to be read out by the DMA Controller via the DMA Fast Message Read register set. Note the DMA Controller must read all 3 DMA Read Registers to obtain the complete message, unlike the case when FIFO is enabled and message is read out from the DMA Fast Message Data Read register only. The data flow will happen as follows:



- A DMA request will be asserted when any of the DMA enabled channels has successfully received one complete Fast Message. The request is kept asserted till all messages (across DMA enabled channels) have been read out.
- DMA Controller should read the complete message by accessing each register of the DMA Fast Message Read register set, the registers being:
  - [DMA Fast Message Data Read Register \(SRX\\_DMA\\_FMSG\\_DATA\)](#)
  - [DMA Fast Message CRC Read Register \(SRX\\_DMA\\_FMSG\\_CRC\)](#)
  - [DMA Fast Message Time Stamp Read Register \(SRX\\_DMA\\_FMSG\\_TS\)](#)
- A fixed round robin sequencing runs, as shown in [Figure 41-82](#), and this logic checks for messages in sequence 0→1→3→5→7 and back to channel 0 and so on. Channel 2 is skipped as it is not DMA enabled) and channel 4 and 6 are skipped as message is not received completely. This sequence continues like this.
- The sequence in which messages are read through DMA interface is not deterministic and depends on the sequence in which they are completely received.
- When the sequencing logic detects a message ready to be read, it will update the DMA Fast Message Read register set for DMA read and move to the next channel when this message is read out completely (i.e., 3 read accesses to the above mentioned registers)
- The DMA controller should be programmed to read only one message in one DMA transfer (i.e. 12 bytes). If it is programmed to read more than one message in one DMA transfer, underflow in buffers might occur
- User software should ensure there is no overflow in channels. In case overflow occurs, the message in the channel buffer will get overwritten by the newly received message. However, when a DMA read is in progress, the buffers will not be overwritten if overflow occurs to maintain data integrity.
- The Round Robin sequencing logic loops through each DMA enabled channel and halts at any channel having data till it is read out by the DMA Controller.

## Functional description

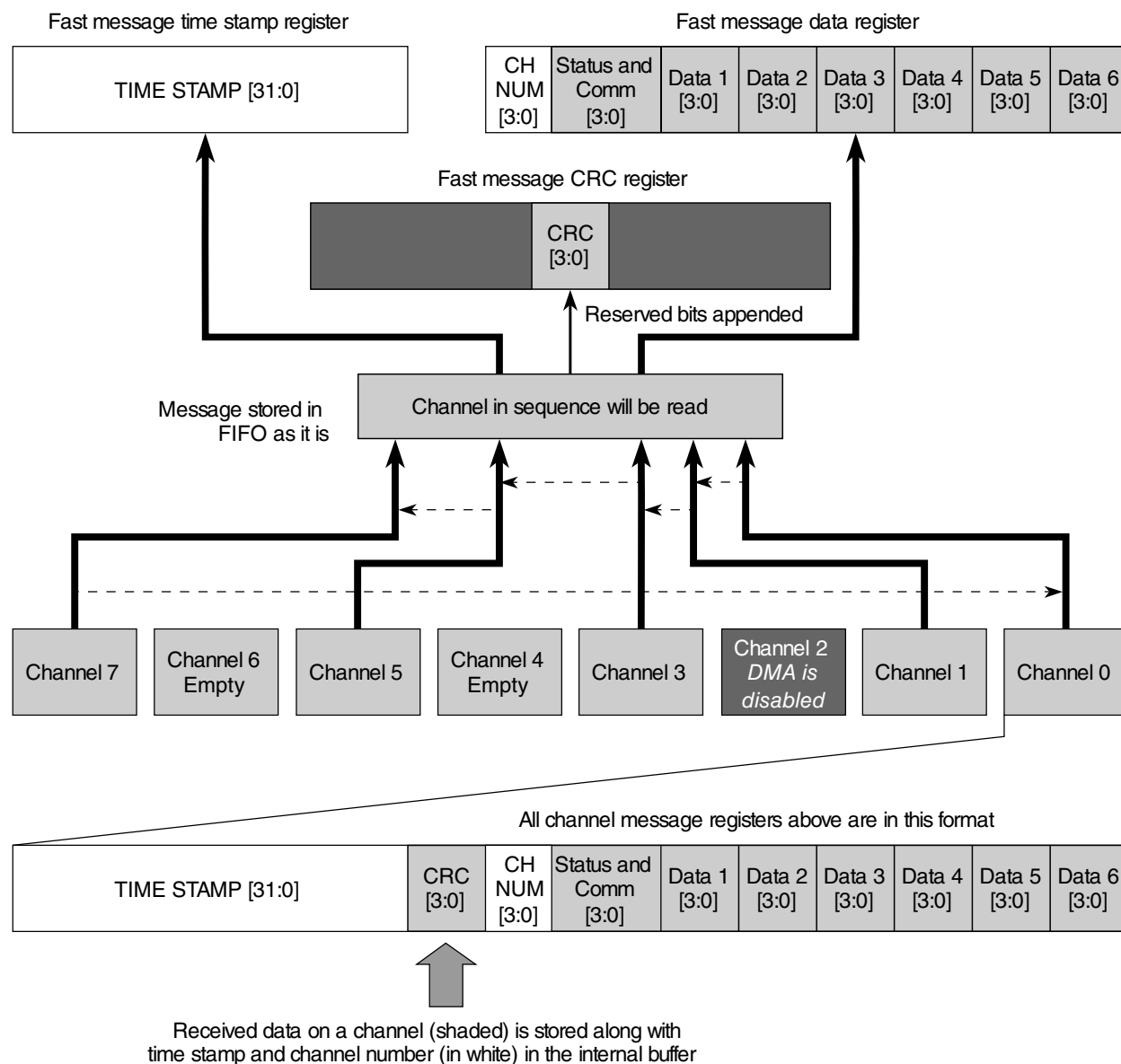


**Figure 41-82. Fast Message DMA read logic (when FIFO is disabled)**

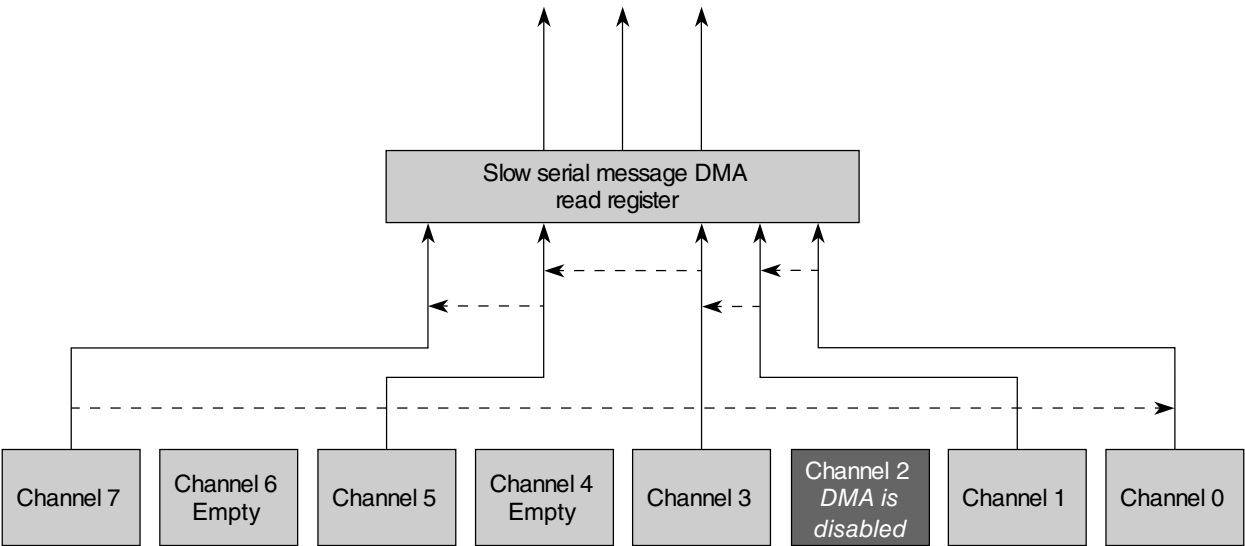
### 41.5.2.3 DMA request for Slow Serial Message reading

A separate DMA request (different from DMA request for Fast Message) is generated for all slow serial messages received on all channels. The data flow will happen as follows:

- Request will be asserted when any of the DMA enabled channels has successfully received one complete serial message (short or enhanced) and kept asserted till all messages have been read from their respective memory-mapped registers

- Similar to DMA read for Fast Messages, data for slow serial messages will be read from DMA Slow Serial Message Read register set in memory map, the register set comprising of:
  - [DMA Slow Serial Message Bit3 Read Register \(SRX\\_DMA\\_SMSG\\_BIT3\)](#)
  - [DMA Slow Serial Message Bit2 Read Register \(SRX\\_DMA\\_SMSG\\_BIT2\)](#)
  - [DMA Slow Serial Message Time Stamp Read Register \(SRX\\_DMA\\_SMSG\\_TS\)](#)
- A fixed round robin sequencing logic will be running and it will update the DMA Slow Serial Message Read register set for DMA read and move to the next channel when this message is read out completely (i.e., 3 DMA read accesses to the above mentioned registers)
- The channel number and time stamp will also be appended for every message
- A message type bit (TYPE) will also be appended to distinguish between short and enhanced serial messages
- The DMA controller should be programmed to read only one message in one DMA transfer (i.e. 12 bytes). If it is programmed to read more than one message in one DMA transfer, underflow in buffers might occur
- Since serial messages receive rate is very slow, user software should ensure there is no overflow
- The fixed round robin sequence logic for DMA read is the same as that used for Fast Messages (see [DMA request for Fast Message reading when FIFO is disabled](#))

[Figure 41-83](#) shows the sequencing of channels for serial messages to be read when DMA read is done on DMA Slow Serial Message Read register set.



**Note:** The arrows in the figure indicate next DMA transfer. One DMA Read Register set (3 registers) will be read in

**Figure 41-83. Slow Serial Message DMA read logic**

The formats in which the 3 types of Serial Messages will be stored in the buffers are shown in figures below.

**Table 41-83. Enhanced Serial Message with 8-bit ID field**

|                   |  |   |    |           |  |  |    |  |   |                   |          |  |   |          |  |   |
|-------------------|--|---|----|-----------|--|--|----|--|---|-------------------|----------|--|---|----------|--|---|
| 31                |  | 1 | 16 |           |  |  | 15 |  | 0 |                   | 0        |  |   |          |  |   |
| CH NUM<br>[3:0]   |  | P |    |           |  |  |    |  | 0 | C                 | ID [7:4] |  | 0 | ID [3:0] |  | 0 |
|                   |  |   |    | CRC [5:0] |  |  |    |  |   | Data Field [11:0] |          |  |   |          |  |   |
| TIME STAMP [31:0] |  |   |    |           |  |  |    |  |   |                   |          |  |   |          |  |   |

**Table 41-84. Enhanced Serial Message with 4-bit ID field**

|                   |  |   |  |           |  |  |  |    |   |                   |          |   |   |                       |  |   |
|-------------------|--|---|--|-----------|--|--|--|----|---|-------------------|----------|---|---|-----------------------|--|---|
| 31                |  | 1 |  | 16        |  |  |  | 15 |   | 1                 |          | 0 |   |                       |  |   |
| CH NUM<br>[3:0]   |  | P |  |           |  |  |  |    | 0 | C                 | ID [3:0] |   | 0 | Data Field<br>[15:12] |  | 0 |
|                   |  |   |  | CRC [5:0] |  |  |  |    |   | Data Field [11:0] |          |   |   |                       |  |   |
| TIME STAMP [31:0] |  |   |  |           |  |  |  |    |   |                   |          |   |   |                       |  |   |

Table 41-83 and Table 41-84 shows the enhanced serial messages. Bit C (shown in figure) or CFG (as shown in DMA Slow Serial Message Bit3 Read Register (DMA\_SMSG\_BIT3) description) defines the type of enhanced packet and bit P (shown in figure) or bit TYPE (shown in DMA Slow Serial Message Bit3 Read Register (DMA\_SMSG\_BIT3) description) defines whether it is short serial or enhanced serial message. CRC, ID and Data fields are stored in format as received and shown in SAE specification with just one exception that CRC is shifted to align with 16-bit boundary.

Table 41-85 shows the short serial message. Field locations have been placed in such a way to allow a common register to read all the slow serial messages and also match the bit positions in actual message defined by SAE Specification.

**Table 41-85. Short serial message**

|                   |   |           |          |                  |
|-------------------|---|-----------|----------|------------------|
| 31                | 0 | 16        | 15       | 0                |
| CH NUM<br>[3:0]   | P |           |          |                  |
|                   |   | CRC [3:0] | ID [3:0] | Data Field [7:0] |
| TIME STAMP [31:0] |   |           |          |                  |

Again, three 32-bit accesses to be made by DMA to get one complete message with time stamp.

### 41.5.3 Message reading via interrupts

There is one interrupt request per channel for a Fast message received and one for a serial message received. Each channel is individually configurable to enable or disable interrupt on reception of Fast or Slow Serial messages separately. In case of interrupt request of Fast Message, an interrupt will be asserted when any of the channels has received a complete Fast Message without errors and that channel is not configured for DMA Reads. The interrupt routine should be coded in such a way to take the least amount of overhead (in CPU cycles) while entering the routine and start reading the available messages. The routine should be capable of accessing all channels that are enabled to give out messages via interrupts through the Register Interface.

In response to these interrupts, the CPU is expected to read interrupt status register for fast or Slow Messages (depending on which interrupt was asserted) and store it in system memory. Using this status register value, the interrupt routine can decide from which channels the message is to be read. For instance if bit corresponding to channel 0 is set then routine should read from the address location corresponding to Channel 0 Fast Message Read registers and move onto the next asserted bit. The interrupt routine should scan through all the bits of this stored interrupt status register value and read corresponding channel's fast or Slow Message registers. In this manner all available messages can be read. After scanning through all the bits of status register stored in system memory, routine can read again the message ready interrupt status register to find out if any other channel has received in the mean time. This will save another interrupt routine time mainly from stacking/unstacking delay.

It is assumed that the interrupt controller will be able to respond fast enough to avoid any overflow as only one message per channel will be stored and the system bus clock frequency is sufficient to avoid this.

### **41.5.3.1 Suggested software interrupt service routine steps**

These are suggested step to have an interrupt routine that can read all messages quickly and avoid any underflow/overflow. Actual implementations can vary.

#### **When separate Fast Message and Slow Serial Message Interrupts are used:**

The Fast Message Interrupt should have higher priority than Slow Serial Message Interrupt. The suggested ISR steps for a Fast Message Interrupts can be

- Read the Fast Message Ready Register and store it in a variable
- Shift out bits starting from MSB from this variable
- If bit is 1, perform read operation on corresponding channel's memory-mapped Fast Message Read Register set (3 registers). Three read accesses will fetch complete message. Shift out new bit. Repeat Step.
- Else if bit is 0, shift out new bit
- When all bits are shifted out, read the Fast Message Status Registers again to ensure no new message has arrived during the ISR.
- If there are no bits set, exit the ISR, else repeat this process.

The ISR steps for Slow Serial Message Interrupt can be same as for Fast Message ISR.

#### **When common Fast Message and Slow Serial Message Interrupts are used:**

The suggested ISR steps can be

- Read the Fast Message Status Register and store it in a variable
- Shift out bits starting from MSB from this variable
- If bit is 1, perform read operation on corresponding channel's memory-mapped Fast Message Read Register. Three read accesses will fetch complete message. Shift out new bit. Repeat Step.
- Else if bit is 0, shift out new bit
- When all bits are shifted out, read the Slow Serial Message Status Register into a new variable

- Shift out bits starting from MSB from this new variable
- If bit is 1, perform read operation on corresponding channel's memory-mapped Slow Serial Message Read Register. Three read accesses will fetch complete message. Shift out new bit. Repeat Step.
- Else if bit is 0, shift out new bit
- When all bits are shifted out, software can again check both Fast and Slow Serial Message Ready register to ready any message that might have been received while the routine was being serviced, or can choose to exit the routine.

## 41.5.4 Overflow behavior

This section describes the overflow behavior.

### 41.5.4.1 Fast messages buffers/FIFO

Overflow in Fast Message buffers or FIFO can occur when the CPU or DMA does not read messages from that channel for a long period of time. The receiver channel stores two messages in its buffers and indicates an overflow when these buffers are full. When FIFO is used, overflow is indicated when the FIFO gets full and another fast message is to be stored.

When an overflow event occurs, the newly received message overwrite the previously stored message and the overflow status bit is set. However, if overflow asserts while CPU/DMA is reading that channel (i.e. all 3 read registers have not yet been read), then that message is not overwritten. While messages are being overwritten, the slow message reception is not hampered.

### 41.5.4.2 Slow message buffers

Overflow in Slow Message buffers can occur when the CPU or DMA does not read messages from that channel for a long period of time. The receiver channel stores two messages in its buffers and indicates an overflow when these buffers are full.

When an overflow event occurs, further reception of slow messages is halted until the buffers are read out by CPU or DMA. Since it is less likely that CPU or DMA do not read the read registers for 32 or 36 fast message length duration, an overflow in slow messages would not be as frequent as in fast messages.

### 41.5.5 Adjustment for variation in sensor (Tx) clock

The clock used by the Channel Receiver (referred to as Receiver Clock, or Rx\_CLK) to sample the message nibbles is generated from a High Frequency Receiver Clock or Protocol Clock. This high frequency receiver clock is divided using a prescaler counter which generates a tick when it rollovers on reaching a count equal to the Prescaler Value programmed in the channel's clock control register. The receive clock is generated separately for each channel. By varying this Prescaler Factor, we can control the period of the receive clock and make it close to the Transmitter clock used by the sensor for the particular channel.

After reset and once channel is enabled, the compensated prescaler value is set to Zero and the user software programs the Prescaler Value to obtain the desired receiver clock frequency for the particular channel. Thus, the receiver starts working by assuming that the transmitter is sending messages on the frequency that is programmed in its channel clock control register. The channel's input after filtering is continuously measured by the Prescaler Counter. The compensation logic checks each pulse length and determines the calibration pulse from them. When a calibration pulse is detected, the compensated prescaler value is computed and stored in the Channel's Clock Control Register ([Channel 'n' Clock Control Register \(n = 0 to \(CH-1\)\) \(SRX\\_CHn\\_CLK\\_CTRL\)](#)). Software can use this value to determine the variation in Tx Clock.

#### Note

A pause pulse may incidentally be detected as a synchronization/calibration pulse during start up after reset. The correction value would be computed on the pause pulse too but since the next synchronization pulse would follow, the logic would again detect the falling edge on the synchronization pulse and the correction value would now be correctly computed on the correct synchronization pulse.

The Clock Compensation or Correction happens for every messages that is received and the compensated prescaler value remains constant for that message.

#### 41.5.5.1 Adjustment for nibble length variation

Section 6.2 of SAE Specifications specifies the maximum allowable limits for clock drift and jitter for the transmitter and receiver separately. This can cause the length of nibbles to vary and can cause incorrect sampling of data nibbles. This variation is adjusted by the SENT Receiver when nibbles are being sampled.



### 41.5.6 Input programmable filter

The Input Programmable Filter ensures that only valid input pin transitions are received by the SENT Receiver module. The input programmable filter is an 8-bit programmable up counter that increments on the high frequency receiver clock. The sensor input signal from device's pad is synchronized by high frequency receiver clock. When a state change occurs in the sensor input signal, the 8-bit counter resets and starts counting up on the high frequency receiver clock. As long as the new state is stable on the pin, the counter remains incrementing. If a counter overflow occurs, the new pin value (i.e. sensor input) is validated and latched in the filtered output flop (also running on high frequency receiver clock). If the opposite edge appears on the sensor input before validation (i.e. overflow), the counter is reset and the input is not passed onto the filtered output. At the next pin transition, the counter starts counting again. Any pulse that is shorter than a full range of the masked counter is regarded as a glitch and it is not passed on to the filter output. The number of samples (or the match value of the up counter) is programmable by the user in the channel's configuration register [Channel 'n' Configuration Register \(n=0 to \(CH-1\)\) \(SRX\\_CHn\\_CONFIG\)](#). When zero is programmed into the registers, it bypasses the filter and the synchronized input is output as the filtered output. A non-zero value delays the filtered output by the number of clocks proportional to the programmed value in the Channel Configuration Register ([Channel 'n' Configuration Register \(n=0 to \(CH-1\)\) \(SRX\\_CHn\\_CONFIG\)](#)). A timing diagram of the input filter when FIL\_CNT is 1, is shown in the following figure.

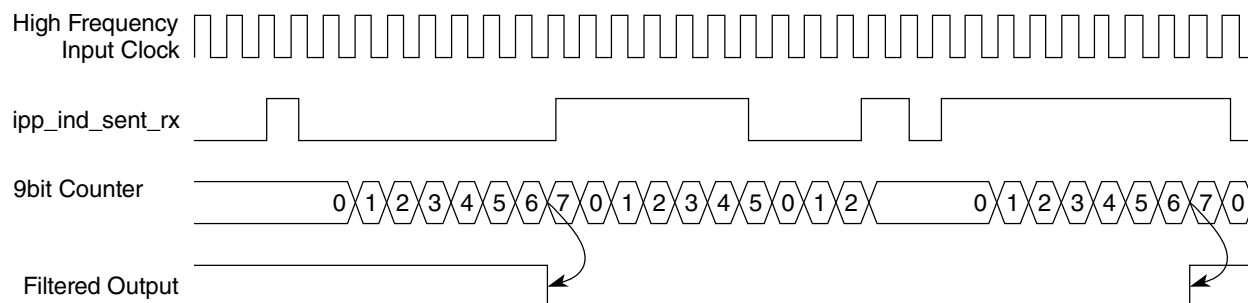


**Figure 41-84. Input programmable filter timing diagram**

### 41.5.7 Receiver diagnostics

Most of the receiver diagnostics are controlled by the Fast Message State Machine. The following checks are conducted by the receiver in each channel

- Calibration pulse length < 56 clock ticks – 25% or > 56 clock ticks + 25%

- Not the expected number of falling edges between calibration pulses. (Message length is pre-defined by each sensor device and programmed by user for each channel)
- Checksum error. Two 4-bit CRC checks and one 6-bit CRC check according to the message type. User has a programmable option to select the method of CRC to use i.e. Legacy or Recommended.
- Any nibble data values measured as  $< 0$  or  $> 15$
- Successive calibration pulses differ by  $> +1.5625\%$  (1/64) or  $< -1.5625\%$ . The accuracy of this check will depend on the frequency of the protocol clock used.
- For messages with pause pulse, an additional diagnostic is done; however, this diagnostic will not fail any message (to reduce latency in message reception) but the failure of this diagnostic will be indicated in the channel's status register.

On detection of any of the above errors, the receiver rejects the current packet until a new calibration pulse is detected. Occurrence of an error is flagged in the respective channel's status register ([Channel 'n' Status Register \(n=0 to \(CH-1\)\) \(SRX\\_CHn\\_STATUS\)](#)). If error interrupt is enabled, an error interrupt will be generated. There is a single error interrupt generated for all errors, so the CPU must read the status register to find out which error occurred. Once error condition is flagged, subsequent errors (except NUM\_EDGES\_ERR) are automatically masked out until a valid calibration pulse is detected, after which all checks are automatically enabled. This is done to avoid multiple assertions of the interrupt to CPU in case the channel become unstable and send wrong data continuously.

#### 41.5.7.1 Calibration pulse length check

The calibration pulse length check happens in conjunction with Clock Compensation Logic since current calibration pulse is detected in this block. The Clock Compensation Logic detects a calibration pulse of length between 42 and 70 tick counts. This check will assert the CAL\_LEN\_ERR status bit on failure.

#### 41.5.7.2 Successive calibration pulse check

The successive calibration pulse check checks if successive calibration pulses differ by  $> +1.5625\%$  (1/64) or  $< -1.5625\%$ . The accuracy of this check will depend on the frequency of the protocol clock used. Specifically, the higher the frequency, the closer the check will be to 1.5625%. So if protocol clock is  $T_{HF\_CLK}$  and channel tick period is

$T_{TX\_CLK}$  then the check will be of  $1.5625\% \pm \{100 \times (4 \times T_{HF\_CLK}) / (42 \times T_{TX\_CLK})\}$ . So it won't allow variance in clock of a channel by  $1.5625\% - \{100 \times (4 \times T_{HF\_CLK}) / (42 \times T_{TX\_CLK})\}$ . This check will assert the CAL\_DIAG\_ERR status bit on failure.

When Option 2 of successive calibration pulse check is enabled, the CAL\_RESYNC bit will be asserted on every third successive error detected by this check.

### 41.5.7.3 Not the expected number of edges check

The number of negative edges between two successive calibration pulses is counted and compared with expected number of edges in case of preferred option of successive calibration pulse check. The expected number of nibbles is the sum of following:

- Status and Communication nibble (1 No.)
- Data nibbles (Programmable as per user software)
- CRC nibble (1 No.)
- Pause Pulse, if configured (1 No.)

This check ("Not the expected number of edges check") will assert the NUM\_EDGES\_ERR status bit on failure. The check does not run when Option 2 (low latency) of successive calibration pulse check is enabled. This check is not masked after any other error is detected so it could be redundant if it is flagged along with CAL\_DIAG\_ERR and CAL\_LEN\_ERR. However, this is true only when CPU is clearing error status bits as and when they are asserted. If errors are accumulated, the above cannot be deduced.

### 41.5.7.4 Nibble value check

As part of this check, the lengths of the following nibbles are checked to remain in between 0 and 15 nibble values(11.5 ticks to 27.5 ticks):

- Status and Communication nibble
- all Data nibbles
- CRC nibble

The nibble value check will assert the NIB\_VAL\_ERR status bit on failure. This error can be flagged along with NUM\_EDGES\_ERR in which case the former can be ignored as the number of pulses in the message is incorrect. However, this is true only when CPU is clearing error status bits as and when they are asserted. In case, errors are accumulated, the above cannot be deduced.

#### **41.5.7.5 CRC check**

The CRC check will compute both 4-bit and 6-bit CRC as requested by State Machine control logic. Two 4-bit CRCs and one 6-bit CRC are supported to compute the CRC on Fast/Slow Serial Message (4-bit CRC) and Enhanced Serial Message (6-bit CRC) respectively. The SENT Receiver Module allows the user to calculate the 4-bit CRC using either the legacy CRC implementation method or the new recommended CRC implementation via a control bit in the channel's corresponding control register. Please refer to SAE SENT Specification for details on CRC algorithm.

Asserting the CHn\_CONFIG\_REG[11] bit, causes the Status and Communication nibble to be included in the 4-bit CRC calculation for Fast Serial Messages. De-asserting this bit enables normal operation as described above.

This check will assert the FMSG\_CRC\_ERR or SMSG\_CRC\_ERR status bit on failure, depending on whether the check is run for Fast Message or Slow Message.

The FMSG\_CRC\_ERR can be flagged along with the NUM\_EDGES\_ERR, in which case the CRC error can be ignored as the number of pulses in the message was incorrect. However, this is true only when CPU is clearing error status bits as and when they are asserted. If errors are accumulated, the above cannot be deduced.

#### **41.5.7.6 Pause pulse diagnostic**

The pause pulse diagnostic checks whether the ratio of calibration pulse length to full message length (including pause pulse) varies from one message to another by less than 1.5625% (as per spec). In order to have reduced latency of operation, message reception is not gated due to this check and no messages are discarded. However, when this check fails, a status bit is asserted to let the software know of this condition. The successive calibration pulse check can be configured to operate along with this check. This configurability is added to avoid conflicting results from the two checks.

There can be some inaccuracy in measuring pause pulse length as SAE specification specifies maximum clock drift/jitter up to maximum nibble length only and the frequency of high frequency receiver clock (protocol clock) used, allows clock compensation logic

to ensure accuracy up to maximum nibble length. Hence there could be some inaccuracy in this check when sampling pause pulses that are larger than the maximum nibble pulse length.

This check will assert the PP\_DIAG\_ERR status bit on failure. This check does not run when reception of pause pulse is disabled.

When the Single Edge Nibble Transmission (SENT) Receiver (SRX) is configured to receive a pause pulse (Channel 'n' Configuration Register - CHn\_CONFIG[PAUSE\_EN] = 1) and the length of the pause pulse from the sensor is in range of the valid calibration pulse length, the NUM\_EDGES error can get asserted spuriously (Channel 'n' Status Register - CHn\_STATUS(NUM\_EDGES\_ERR) = 1) when there is any diagnostic error (other than number of expected edges error) or overflow in the incoming messages from the sensor.

Software can distinguish a spurious NUM\_EDGES\_ERR error from a real one by monitoring other error bits. The following tables will help distinguish between a false and real assertion of NUM\_EDGES\_ERR error and other errors. Software should handle the first error detected as per application needs and other bits can be evaluated based on these tables. Table 1 contains information due to erratum behavior and Table 2 below contains clarification of normal NUM\_EDGES\_ERR behavior.

**Table 41-86. Behavior of NUM\_EDGES\_ERR**

| First Error Detected | Other error bits asserted    | Cause for extra error bits getting asserted                                                                                                                                                                                                                                                                                                                                     | Action                           |
|----------------------|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------|
| NIB_VAL_ERR          | NUM_EDGES_ERR asserted twice | Upon detection of the first error, the state machine goes into a state where it waits for a calibration pulse, the first NUM_EDGES_ERR error is for the current message as the state machine does not detect an end of message. The second error comes when both the Pause pulse and the Calibration pulse are seen as back to back calibration pulses and no edges in between. | Ignore both NUM_EDGES_ERR error  |
| FMSG_CRC_ERR         | NUM_EDGES_ERR asserted twice | Upon detection of the first error, the state machine goes into a state where it waits for a calibration pulse, the first NUM_EDGES_ERR error is for the current message as the state machine does not detect an end of message. The second error comes when both the Pause pulse and the Calibration pulse are                                                                  | Ignore both NUM_EDGES_ERR errors |

*Table continues on the next page...*

**Table 41-86. Behavior of NUM\_EDGES\_ERR  
(continued)**

|             |                             |                                                                                                                                                                                                                                      |                            |
|-------------|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------|
|             |                             | seen as back to back calibration pulses and no edges in between.                                                                                                                                                                     |                            |
| CAL_LEN_ERR | NUM_EDGES_ERR asserted once | Since the calibration pulse is not detected as a valid calibration pulse, the internal edges counter does not detect the end of one message and start of bad message (which has CAL_LEN_ERR); hence the NUM_EDGES_ERR gets asserted. | Ignore NUM_EDGES_ERR error |

Resynchronization of SENT Receiver in case of any diagnostic error is shown in figure [Figure 41-85](#).

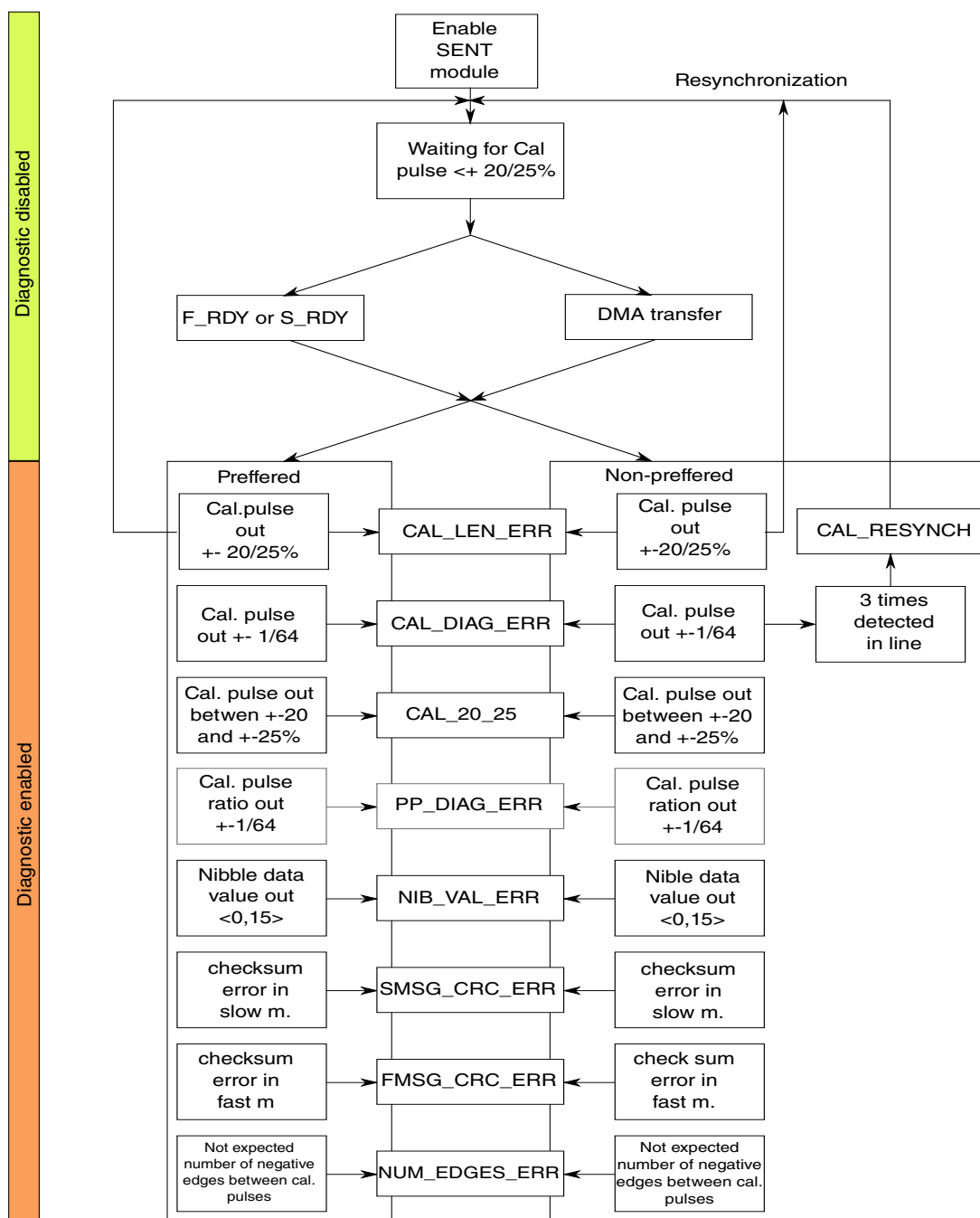


Figure 41-85. SENT receiver resynchronization

Table 41-87. Expected behavior, clarification of NUM\_EDGES\_ERR cases

| First Error Detected                              | Other error bits asserted | Cause for extra error bits getting asserted                                                                                                                 | Action                 |
|---------------------------------------------------|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|
| NUM_EDGES_ERR (when edges are less than expected) | NIB_VAL_ERR is asserted   | When the actual number of edges in the message are less than expected, then a pause pulse gets detected as a nibble since the state machine expects nibbles | Ignore the NIB_VAL_ERR |

Table continues on the next page...

**Table 41-87. Expected behavior, clarification of NUM\_EDGES\_ERR cases (continued)**

|                                                   |                                          |                                                                                                                                                                                                                                                                                                                                                        |                                    |
|---------------------------------------------------|------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------|
|                                                   |                                          | when actually there is a pause pulse present. This generates NIB_VAL_ERR.                                                                                                                                                                                                                                                                              |                                    |
| NUM_EDGES_ERR (when edges are more than expected) | NIB_VAL_ERR and PP_DIAG_ERR are asserted | When the actual number of edges in a message are more than expected, then after receiving the programmed number of data nibbles, the state machine expects a pause pulse. However, the pause pulse comes later and gets detected as a nibble and hence NIB_VAL_ERR is asserted. Since the message length is not correct, PP_DIAG_ERR is also asserted. | Ignore NIB_VAL_ERR and PP_DIAG_ERR |

### 41.5.8 Time stamp logic

A single counter maintains the time stamp for all channels which can be used by the software to determine the relative time of arrival of the messages for each channel. The time stamp counter is a 32-bit counter.

The clock for this counter is derived from the high frequency receiver clock using a Time Stamp Prescaler Counter that divides the high frequency receiver clock to generate the clock of required resolution. The user must program the time stamp prescaler value (in [Global Control Register \(SRX\\_GBL\\_CTRL\)](#)) in order to generate the time stamp clock. The period of the time stamp clock must be less than that of the fastest receiver channel being used, or it can be set at some fixed value determined by application, such as 1  $\mu$ s. For example, if the high frequency receiver clock is 60 MHz, the required prescaler value is 59 to generate a time stamp clock of 1  $\mu$ s resolution.

#### Note

The rollover of time stamp values should be determined in user software, by comparing the time stamp values in previous and current messages that are read out. User software should adjust the time stamp values in these messages accordingly, after reading them.



The time stamp for a Fast Message is sampled at the falling edge that comes at the end of the calibration pulse (which is also the start of the status and communication nibble pulse) and is stored in the message read buffer. However, if the current message is found to be in error, the message buffer is discarded. Thus, the time stamp will be available to user software for only those messages which do not have any error.

The time stamp for a serial message is sampled at the end of the message i.e. when the serial message is completely received and when the message is found to have no errors. the time stamp is stored in the message read buffer to be read by the user software. Again, for messages with error the message and time stamp are discarded.

#### 41.5.8.1 Limitation

Since the same counter is used for all channels, it may happen that more than 1 message is received within the same time stamp across different channels. Hence the software would not be able to determine the sequence of arrival of messages which have the same time stamp value.

### 41.5.9 Bus Idle Diagnostic

This diagnostic checks the connected sensors for inactivity. SENT module can indicate if the idle period on a particular channel has crossed the period defined by the programmed value in Bus Idle Count (in [Channel 'n' Configuration Register \(n=0 to \(CH-1\)\) \(SRX\\_CHn\\_CONFIG\)](#)). Status bit (bus\_idle in [Channel 'n' Status Register \(n=0 to \(CH-1\)\) \(SRX\\_CHn\\_STATUS\)](#)) will be set when idle period crosses the allowed value. CPU should write 1 to this clear this bit once it is read.

## 41.6 Clocks and resets

The SENT Receiver works on two clocks. First is the System Bus Clock that is used to program the registers and read messages via DMA or Interrupt from the module's Register Interface. Second is the High Frequency Receiver Clock or Protocol Clock that is used for accurate message receiving operations. There is a single asynchronous reset which is the system asynchronous reset and one internal reset for each of the channel logic.

## 41.6.1 Clocking strategy

The following figure shows the clock domain in which each module is functioning.

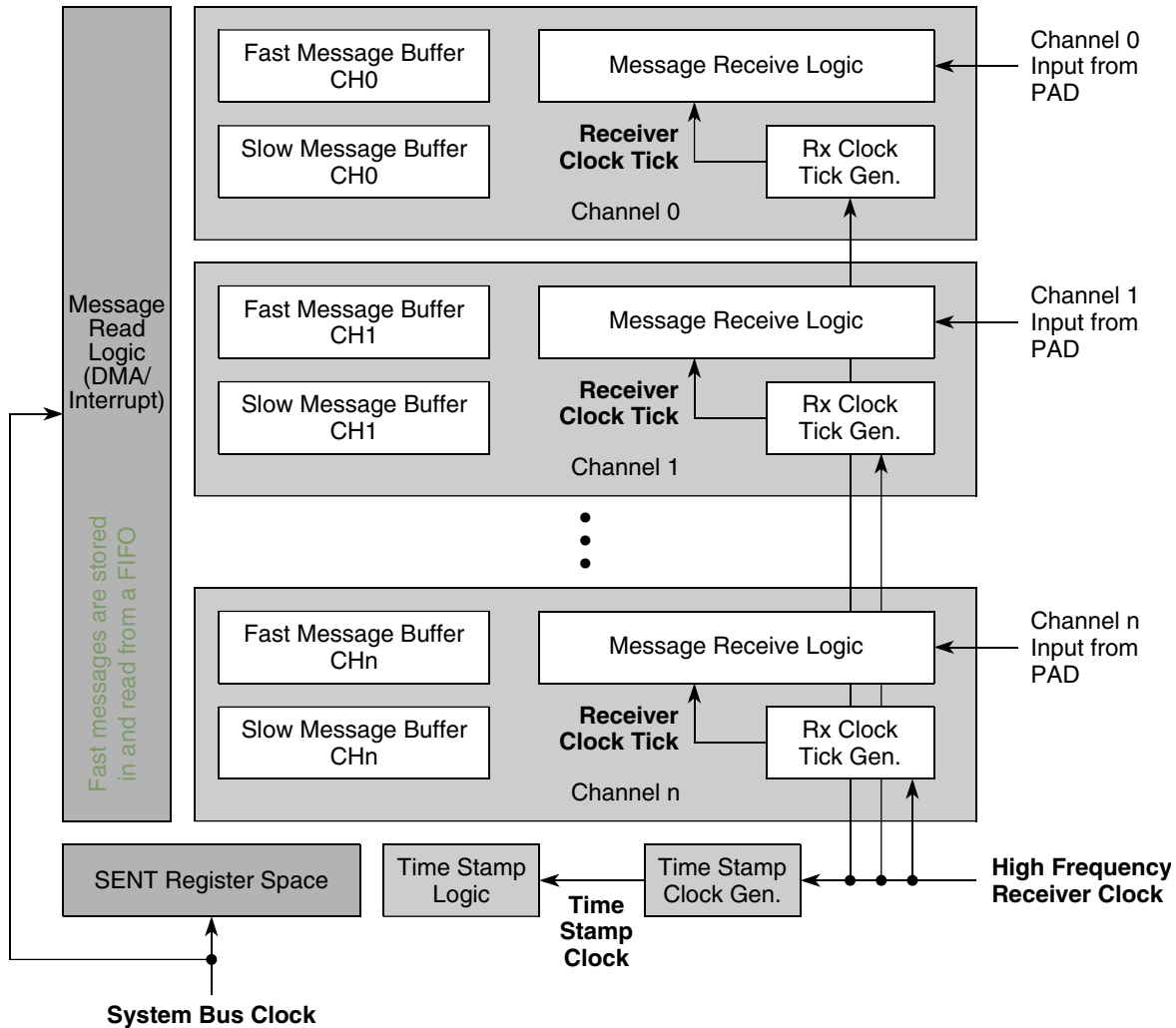


Figure 41-86. SENT Receiver module clock domains

## 41.6.2 System bus clock requirements

Since all the received messages are read on the System Bus Clock it is necessary that both the interrupt service routine and the DMA controller must be able to access all channels with messages ready to be read within a time to prevent overflow in any channel. The minimum time available to the CPU or DMA is the case when the smallest Fast Message is received back to back on all channels while working on the smallest receive clock tick period.

Now, the smallest Fast Message size with valid checksum as per SAE Specifications is 154 ticks. At 3  $\mu$ s receive clock tick with a -20% variation in the transmitter clock tick, the effective receiver clock tick is 2.4  $\mu$ s (after compensation). This gives us the time in which the minimum size Fast Message is received in any channel as  $154 \times 2.4 = 369.6$   $\mu$ s. Now since messages are arriving in parallel, 369.6  $\mu$ s is the time available for CPU or DMA to access all channels to prevent an overflow in the channel that received the very first message.

Thus, if assume that ' $N_{\text{CHNL}}$ ' channels are being used and all channels have messages ready to be read together from either DMA or CPU then the time available to read a message in one channel will be  $369.6/N_{\text{CHNL}}$   $\mu$ s or  $\sim 370/N_{\text{CHNL}}$   $\mu$ s. Since the CPU or the DMA needs to make three 32-bit accesses in order to read a complete message and taking the time required for one 32-bit read access as ' $T_{\text{RD}}$ ' (assuming it to be same for DMA and CPU); we can determine the minimum frequency for the System Bus Clock as:

For CPU,

$$3 \times T_{\text{RD}} + T_{\text{ISR}} + T_{\text{WAKUP}} < 370 / N_{\text{CHNL}}$$

where  $T_{\text{ISR}}$  is the overhead in initiating an ISR and  $T_{\text{WAKUP}}$  is the time to wakeup from low power mode

For DMA,

$$3 \times T_{\text{RD}} + T_{\text{WAKUP}} < 370 / N_{\text{CHNL}}$$

Replacing  $T_{\text{RD}} = N_{\text{RD}} \times T_{\text{SYS\_CLK}}$  and  $T_{\text{ISR}} = N_{\text{ISR}} \times T_{\text{SYS\_CLK}}$ , where  $N_{\text{RD}}$  is the number of System Bus Clock cycles in one 32-bit read access,  $N_{\text{ISR}}$  is the number of System Bus Clock Cycles in initiating an interrupt service routine and  $T_{\text{SYS\_CLK}}$  is the period of the same clock, we get

For CPU,

$$(3 \times N_{\text{RD}} + N_{\text{ISR}}) \times T_{\text{SYS\_CLK}} + T_{\text{WAKUP}} < 370 / N_{\text{CHNL}}$$

(all units is  $\mu$ s)

For DMA,

$$3 \times N_{\text{RD}} \times T_{\text{SYS\_CLK}} + T_{\text{WAKUP}} < 370 / N_{\text{CHNL}}$$

(all units is  $\mu$ s)

### 41.6.3 High frequency receiver clock (protocol clock) requirements

For accurate receive operations, the protocol is required to satisfy a minimum frequency defined as per formula:  $\text{Guard Band} > 2.5 * T_{\text{HF\_CLK}}$

where  $T_{\text{HF\_CLK}}$  is the High Frequency Receiver Clock or Protocol Clock in ns and Guard Band is guard band in ns as specified in SAE specification, at particular transmitter uTick.

#### Note

$T_{\text{HF\_CLK}}$  in the above equation should be chosen based on the fastest sensor clock (with minimum guard band as specified in SAE spec) among all channels in the device.

# Chapter 42

## Enhanced Serial Communication Interface (eSCI)

### 42.1 Chip-specific eSCI information

This chip has six instances of the eSCI module. Some feature details vary among the instances.

The following table summarizes the feature differences. The table does not list feature details that the instances share.

**Table 42-1. eSCI instance feature differences**

| Instance                           | DMA support                                                                |
|------------------------------------|----------------------------------------------------------------------------|
| eSCI_A and eSCI_B                  | Yes                                                                        |
| eSCI_C, eSCI_D, eSCI_E, and eSCI_F | No: descriptions of eSCI DMA functionality do not apply to these instances |

#### NOTE

For eSCI\_D, the single wire feature does not apply for TX/RX via PCSA3 because this pad works only as an output.

#### 42.1.1 eSCI clocking

For information about eSCI clock sources and the eSCI's module clock (MCLK), see the [Clock generation](#) diagram, the [Peripheral clocks](#) diagram, and other details in the Clocking chapter.

#### 42.1.2 eSCI registers: core access in supervisor mode

On this chip, all core accesses to eSCI registers occur in supervisor mode.

## 42.2 Introduction

The eSCI block is an enhanced SCI block with a LIN master interface layer and DMA support. The LIN master layer complies with the specifications LIN 1.3, LIN 2.0, LIN 2.1, and SAE J2602/1.

### 42.2.1 Bibliography

- LIN Specification Package Revision 1.3; December 12, 2002
- LIN Specification Package Revision 2.0; September 23, 2003
- LIN Network for Vehicle Applications, SAE J2602/1, September 1, 2005
- LIN Specification Package Revision 2.1; November 24, 2006

### 42.2.2 Acronyms and Abbreviations

The following table contains acronyms and abbreviations used in this document.

**Table 42-2. Acronyms and Abbreviations**

| Term   | Description                                                                     |
|--------|---------------------------------------------------------------------------------|
| eSCI   | Enhanced SCI block with LIN support and DMA support                             |
| SCI    | Serial Communications Interface                                                 |
| LIN    | Local Interconnect Network - A protocol for low-cost automobile networks        |
| LIN PE | LIN Protocol Engine, Finite State Machine to control logic of the LIN hardware. |
| MCLK   | Module Clock, defined in <a href="#">Module Clock</a>                           |
| TCLK   | Transmitter Clock, defined in <a href="#">Transmitter Clock</a>                 |
| RCLK   | Receiver Clock, defined in <a href="#">Receiver Clock</a>                       |
| RSC    | Receiver Sample Counter, defined in <a href="#">Receiver Clock</a>              |

### 42.2.3 Glossary

**Table 42-3. Glossary**

| Term            | Definition                                              |
|-----------------|---------------------------------------------------------|
| Logic level one | The voltage that corresponds to Boolean true (1) state. |

*Table continues on the next page...*

**Table 42-3. Glossary (continued)**

| Term             | Definition                                                                                                                                                                                                         |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Logic level zero | The voltage that corresponds to Boolean false (0) state.                                                                                                                                                           |
| Set              | To set a bit or bits means to establish logic level one on the bit or bits.                                                                                                                                        |
| Clear            | To clear a bit or bits means to establish logic level zero on the bit or bits.                                                                                                                                     |
| Asserted         | A signal that is asserted is in its active state. An active low signal changes from logic level one to logic level zero when asserted, and an active high signal changes from logic level zero to logic level one. |
| Preamble         | The term preamble describes an idle character which is <i>transmitted</i> by the eSCI module.                                                                                                                      |
| Bit time         | Duration of a single bit in a transmitted byte field or character, equivalent to the duration of one transmitter clock cycle defined in <a href="#">Transmitter Clock</a>                                          |
| frame            | Entity that consists of the start bit followed by payload bits followed by one or more stop bits                                                                                                                   |
| LIN byte field   | Special instance of a frame                                                                                                                                                                                        |
| SCI frame        | Special instance of a frame                                                                                                                                                                                        |
| LIN frame        | Sequence of break character followed by LIN byte fields                                                                                                                                                            |
| LIN TX frame     | A LIN frame with the frame header, data byte fields, and checksum field transmitted by the eSCI module                                                                                                             |
| LIN RX frame     | A LIN frame with the header field transmitted by the eSCI module and the data byte fields and checksum field received by the eSCI module                                                                           |
| module is idle   | Module is idle, described in <a href="#">Module Idle Condition</a>                                                                                                                                                 |

## 42.2.4 Overview

The eSCI block allows asynchronous serial communications with peripheral devices and other CPUs. It includes special support to interface to LIN slave devices.

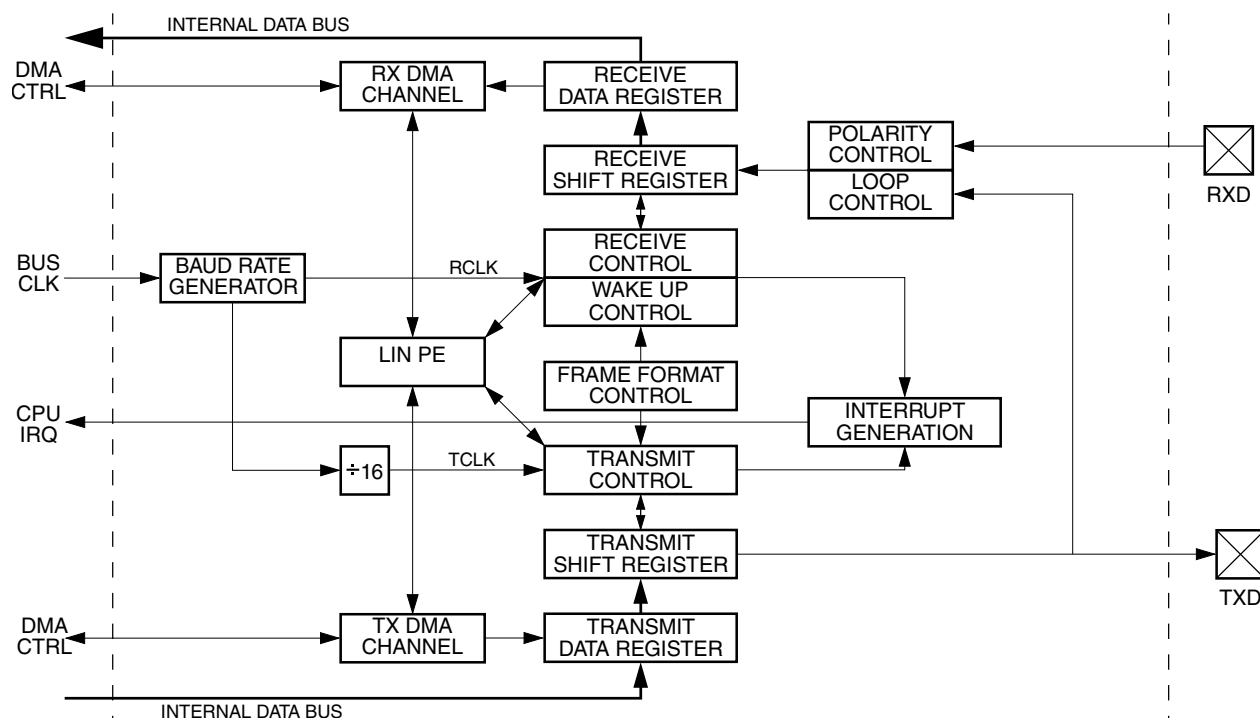


Figure 42-1. eSCI Block Diagram

## 42.2.5 Features

The eSCI block includes these distinctive features:

- Full-duplex operation
- Standard mark/space non-return-to-zero (NRZ) format
- 13-bit baud rate selection
- Programmable frame, payload, and character format
- Support of 2 stop bits in receiver path
- Hardware parity generation and checking
  - Programmable even or odd parity
- Programmable polarity of RXD pin
- Separately enabled transmitter and receiver
- Two receiver wake-up methods:



- Idle line wake up
- Address mark wake up
- Interrupt-driven operation with eight flags:
  - Transmitter empty
  - Transmission complete
  - Receiver full
  - Idle receiver input
  - Receiver overrun
  - Noise error
  - Framing error
  - Parity error
- Receiver framing error detection
- 1/16 bit-time noise detection
- 2 channel DMA interface
- LIN support
  - LIN Master Node functionality (master and slave task)
  - Compatible with LIN slaves from revisions 1.x and 2.0 of the LIN standard
  - Detection of Bit Errors, Physical Bus Errors and Checksum Errors
  - All status bit can generate maskable interrupts
  - Application layer CRC support
  - Programmable CRC polynomial
  - Detection and generation of wake-up characters
  - Programmable wake-up delimiter time
  - Programmable slave timeout
  - Can be configured to include header bits in checksum
  - LIN DMA interface

## 42.2.6 Modes of Operation

The eSCI module has two functional operational modes, SCI and LIN mode. The availability of register bits and fields depends on the selected operational mode.

### 42.2.6.1 Module Idle Condition

Doze mode, Disable mode and Stop mode can only be entered when the module is idle. The module is idle if

- all five active status bits DACT, WACT, LACT, TACT, and RACT in the Interrupt Flag and Status Register 1 (eSCI\_IFSR1) are 0, and
- no interrupt request is pending, i.e either the interrupt flag or its related interrupt enable is 0.

To ensure that the module goes idle, the application should clear all interrupt enable bits before triggering the mode change.

### 42.2.6.2 SCI Mode

The SCI mode is the default functional operational mode and is described in [SCI Mode](#).

### 42.2.6.3 LIN Mode

The LIN mode is the second functional operational mode and is described in [LIN Mode](#).

### 42.2.6.4 Disabled Mode

In the Disabled mode the eSCI module indicates to the clocking system, that all module clocks can be turned off.

The eSCI module is in the Disabled Mode, if the MDIS bit in the Control Register 2 (eSCI\_CR2) is set and the module is idle.

## 42.3 External Signal Description

The eSCI module is connected to a total of two external pins.

### 42.3.1 Detailed Signal Descriptions

#### 42.3.1.1 eSCI Transmit Pin (TXD)

This pin serves as transmit data output and as the receive data input of eSCI.

#### 42.3.1.2 eSCI Receive Pin (RXD)

This pin serves as receive data input of the eSCI.

## 42.4 Memory Map and Register Definition

Writes to a reserved register location do not have any effect and reads of these locations return a zero.

### NOTE

All R/W registers are writable only in supervisor mode of the core performing the write.

**eSCI memory map**

| Address offset (hex) | Register name                                     | Width (in bits) | Access | Reset value | Section/page                |
|----------------------|---------------------------------------------------|-----------------|--------|-------------|-----------------------------|
| 0                    | Baud Rate Register (eSCI_BRR)                     | 16              | R/W    | 0004h       | <a href="#">42.4.1/2112</a> |
| 2                    | Control Register 1 (eSCI_CR1)                     | 16              | R/W    | 0000h       | <a href="#">42.4.2/2113</a> |
| 4                    | Control Register 2 (eSCI_CR2)                     | 16              | R/W    | 2000h       | <a href="#">42.4.3/2115</a> |
| 6                    | SCI Data Register (eSCI_SDR)                      | 16              | R/W    | 0000h       | <a href="#">42.4.4/2117</a> |
| 8                    | Interrupt Flag and Status Register 1 (eSCI_IFSR1) | 16              | w1c    | 8000h       | <a href="#">42.4.5/2119</a> |
| A                    | Interrupt Flag and Status Register 2 (eSCI_IFSR2) | 16              | w1c    | 4000h       | <a href="#">42.4.6/2121</a> |
| C                    | LIN Control Register 1 (eSCI_LCR1)                | 16              | R/W    | 0000h       | <a href="#">42.4.7/2122</a> |
| E                    | LIN Control Register 2 (eSCI_LCR2)                | 16              | R/W    | 0000h       | <a href="#">42.4.8/2124</a> |

*Table continues on the next page...*

## eSCI memory map (continued)

| Address offset (hex) | Register name                          | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|----------------------------------------|-----------------|--------|-------------|------------------------------|
| 10                   | LIN Transmit Register (eSCI_LTR)       | 8               | R/W    | 00h         | <a href="#">42.4.9/2124</a>  |
| 14                   | LIN Receive Register (eSCI_LRR)        | 8               | R      | 00h         | <a href="#">42.4.10/2126</a> |
| 18                   | LIN CRC Polynomial Register (eSCI_LPR) | 16              | R/W    | C599h       | <a href="#">42.4.11/2127</a> |
| 1A                   | Control Register 3 (eSCI_CR3)          | 16              | R/W    | 0000h       | <a href="#">42.4.12/2127</a> |

### 42.4.1 Baud Rate Register (eSCI\_BRR)

This register provides the control value for the serial baud rate. The baud rate and clock generation is specified in [Baud Rate and Clock Generation](#).

A byte write access to only the upper byte of this register (eSCI\_BRR[0:7 ]) will not change the content of the register, instead, the written byte is stored internally into a shadow register. A subsequent byte write access to only the lower byte of this register (eSCI\_BRR[8:15 ]) updates the lower byte and copies the content of the shadow register into the upper byte.

A byte write access to only the lower byte of this register (eSCI\_BRR[8:15 ]) without a preceding byte write access to only the upper byte copies a value of all zero into the upper byte.

A 16-bit write access to this register updates both the lower and upper byte immediately and is the recommended write access type for this register.

Address: 0h base + 0h offset = 0h

|       |   |   |   |     |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|---|---|---|-----|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3   | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| Read  | 0 |   |   | SBR |   |   |   |   |   |   |    |    |    |    |    |    |
| Write |   |   |   |     |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 1  | 0  | 0  |

#### eSCI\_BRR field descriptions

| Field           | Description                                                                             |
|-----------------|-----------------------------------------------------------------------------------------|
| 0–2<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 3–15<br>SBR     | Serial Baud Rate. This field provides the baud rate control value SBR.                  |

## 42.4.2 Control Register 1 (eSCI\_CR1)

This register provides bits to configure the functionality of the module, provides the interrupt enable bits for the interrupt flags provided in [Interrupt Flag and Status Register 1 \(eSCI\\_IFSR1\)](#) and provides the control bits for the transmitter and receiver.

**Table 42-7. Receive Source Mode Selection**

| LOOPS | RSRC | Receiver Input Mode |
|-------|------|---------------------|
| 0     | 0    | Dual Wire Mode      |
| 0     | 1    | Reserved            |
| 1     | 0    | Loop Mode           |
| 1     | 1    | Single Wire Mode    |

Address: 0h base + 2h offset = 2h

|       |       |          |      |   |      |   |    |    |
|-------|-------|----------|------|---|------|---|----|----|
| Bit   | 0     | 1        | 2    | 3 | 4    | 5 | 6  | 7  |
| Read  | LOOPS | Reserved | RSRC | M | WAKE | 0 | PE | PT |
| Write |       |          |      |   |      |   |    |    |
| Reset | 0     | 0        | 0    | 0 | 0    | 0 | 0  | 0  |

|       |     |      |     |      |    |    |     |     |
|-------|-----|------|-----|------|----|----|-----|-----|
| Bit   | 8   | 9    | 10  | 11   | 12 | 13 | 14  | 15  |
| Read  | TIE | TCIE | RIE | ILIE | TE | RE | RWU | SBK |
| Write |     |      |     |      |    |    |     |     |
| Reset | 0   | 0    | 0   | 0    | 0  | 0  | 0   | 0   |

### eSCI\_CR1 field descriptions

| Field         | Description                                                                                                                                                                                                                                             |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>LOOPS    | Loop Mode Select. This control bit together with the RSRC control bit defines the receiver source mode. The mode coding is defined in <a href="#">Table 42-7</a> and the modes are described in <a href="#">Receiver Input Mode Selection</a> .         |
| 1<br>Reserved | This field is reserved.<br>Application must not write 1 to this bit.                                                                                                                                                                                    |
| 2<br>RSRC     | Receiver Source Control. This control bit together with the LOOPS control bit defines the receiver source mode. The mode coding is defined in <a href="#">Table 42-7</a> and the modes are described in <a href="#">Receiver Input Mode Selection</a> . |
| 3<br>M        | Frame Format Mode. This control bit together with the M2 bit of the <a href="#">Control Register 3 (eSCI_CR3)</a> controls the frame format used. The supported frame formats and the related settings are defined in <a href="#">Frame Formats</a> .   |
| 4<br>WAKE     | Receiver Wake-up Condition. This control bit defines the wake-up condition for the receiver. The receiver wake-up is described in <a href="#">Multiprocessor Communication</a> .<br>0 Idle line wake-up.<br>1 Address mark wake-up                      |
| 5<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                 |
| 6<br>PE       | Parity Enable. This control bit enables the parity bit generation and checking. The location of the parity bits is shown in <a href="#">Frame Formats</a> .                                                                                             |

*Table continues on the next page...*

## eSCI\_CR1 field descriptions (continued)

| Field      | Description                                                                                                                                                                                                                                                                                                  |
|------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|            | 0 Parity bit generation and checking disabled.<br>1 Parity bit generation and checking enabled.                                                                                                                                                                                                              |
| 7<br>PT    | Parity Type. This control bit defines whether even or odd parity has to be used.<br><br>0 Even parity (even number of ones in character clears the parity bit).<br>1 Odd parity (odd number of ones in character clears the parity bit).                                                                     |
| 8<br>TIE   | Transmitter Interrupt Enable. This bit controls the eSCI_IFSR1[TRDE] interrupt request generation.<br><br>0 TDRE interrupt request generation disabled.<br>1 TDRE interrupt request generation enabled.                                                                                                      |
| 9<br>TCIE  | Transmission Complete Interrupt Enable. This bit controls the eSCI_IFSR1[TC] interrupt request generation.<br><br>0 TC interrupt request generation disabled.<br>1 TC interrupt request generation enabled.                                                                                                  |
| 10<br>RIE  | Receiver Full Interrupt Enable. This bit controls the eSCI_IFSR1[RDRF] interrupt request generation.<br><br>0 RDRF interrupt request generation disabled.<br>1 RDRF interrupt request generation enabled.                                                                                                    |
| 11<br>ILIE | Idle Line Interrupt Enable. This bit controls the eSCI_IFSR1[IDLE] interrupt request generation.<br><br>0 IDLE interrupt request generation disabled.<br>1 IDLE interrupt request generation enabled.                                                                                                        |
| 12<br>TE   | Transmitter Enable. This control bit enables and disables the transmitter. The control features of the transmitter are described in <a href="#">Transmitter States and Transitions</a> .<br><br>0 Transmitter disabled.<br>1 Transmitter enabled.                                                            |
| 13<br>RE   | Receiver Enable. This control bit enables and disables the receiver. The control features of the receiver are described in <a href="#">Receiver States and Transitions</a> .<br><br>0 Receiver disabled.<br>1 Receiver enabled.                                                                              |
| 14<br>RWU  | Receiver Wake-Up Mode. This bit controls and indicates the receiver wake-up mode, which is described in <a href="#">Multiprocessor Communication</a> .<br><br><b>NOTE:</b> This bit should be set in SCI mode only.<br><br>0 Normal receiver operation.<br>1 Receiver is in wake-up mode.                    |
| 15<br>SBK  | Send Break Character. This bit controls the transmission of break characters, which is described in <a href="#">Break Character Transmission</a> .<br><br><b>NOTE:</b> This bit should be set in SCI mode only.<br><br>0 No break characters will be transmitted.<br>1 Break characters will be transmitted. |

### 42.4.3 Control Register 2 (eSCI\_CR2)

This register provides bits to configure the functionality of the module, and interrupt enable bits for the interrupt flags provided in [Interrupt Flag and Status Register 1 \(eSCI\\_IFSR1\)](#) and control bits for the transmitter and receiver.

Address: 0h base + 4h offset = 4h

| Bit   | 0    | 1   | 2    | 3      | 4     | 5     | 6    | 7     |
|-------|------|-----|------|--------|-------|-------|------|-------|
| Read  | MDIS | FBR | BSTP | BERRIE | RXDMA | TXDMA | BRCL | TXDIR |
| Write |      |     |      |        |       |       |      |       |
| Reset | 0    | 0   | 1    | 0      | 0     | 0     | 0    | 0     |

| Bit   | 8    | 9     | 10    | 11   | 12   | 13   | 14   | 15   |
|-------|------|-------|-------|------|------|------|------|------|
| Read  | BESM | BESTP | RXPOL | PMSK | ORIE | NFIE | FEIE | PFIE |
| Write |      |       |       |      |      |      |      |      |
| Reset | 0    | 0     | 0     | 0    | 0    | 0    | 0    | 0    |

#### eSCI\_CR2 field descriptions

| Field       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>MDIS   | Module Disabled Mode. This bit controls the Module Mode of Operation, which is described in <a href="#">Modes of Operation</a> .<br><br>0 Module is not in Disabled Mode.<br>1 Module is in Disabled Mode, if module is idle.                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 1<br>FBR    | Fast Bit Error Detection. This bit controls the Bit Error Detection mode.<br><b>NOTE:</b> This bit is used in LIN mode only.<br><br>0 Standard Bit error detection performed as described in <a href="#">Standard Bit Error Detection</a> .<br>1 Fast Bit error detection performed as described in <a href="#">Fast Bit Error Detection</a> .                                                                                                                                                                                                                                                                                                                 |
| 2<br>BSTP   | DMA Stop on Bit Error or Physical Bus Error. This bit controls the transmit DMA requests generation in case of bit errors or physical bus errors. Bit errors are indicated by the BERR flag in the <a href="#">Interrupt Flag and Status Register 1 (eSCI_IFSR1)</a> and physical bus errors are indicated by the PBERR flag in the <a href="#">Interrupt Flag and Status Register 2 (eSCI_IFSR2)</a> .<br><b>NOTE:</b> This bit is used in LIN mode only.<br><br>0 Transmit DMA requests generated regardless of bit errors or physical bus errors.<br>1 Transmit DMA requests are not generated if eSCI_IFSR1[BERR] flag or eSCI_IFSR2[PBERR] flags are set. |
| 3<br>BERRIE | Bit Error Interrupt Enable. This bit controls the BERR interrupt request generation.<br><br>0 BERR interrupt request generation disabled.<br>1 BERR interrupt request generation enabled.                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 4<br>RXDMA  | Receive DMA Control. This bit enables the receive DMA feature. When this bit is cleared, a pending receive DMA request is deasserted.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

*Table continues on the next page...*

**eSCI\_CR2 field descriptions (continued)**

| Field       | Description                                                                                                                                                                                                                                                                                                                                   |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|             | 0 Receive DMA disabled.<br>1 Receive DMA enabled.                                                                                                                                                                                                                                                                                             |
| 5<br>TXDMA  | Transmit DMA Control. This bit enables the transmit DMA feature. When this bit is cleared, a pending transmit DMA request is deasserted.<br><br>0 Transmit DMA disabled.<br>1 Transmit DMA enabled.                                                                                                                                           |
| 6<br>BRCL   | Break Character Length. This bit is used to define the length of the break character to be transmitted. The settings are specified in <a href="#">Break Character Formats</a> .                                                                                                                                                               |
| 7<br>TXDIR  | TXD pin output driver control. This bit has no effect. In this device, TXDIR does not control the output driver of the TXD pin in Single Wire Mode. The function is controlled by the relevant Pad Control Register in the System Integration Unit.                                                                                           |
| 8<br>BESM   | Fast Bit Error Detection Sample Mode. This bit defines the sample point for the Fast Bit Error Detection Mode.<br><br><b>NOTE:</b> This bit is used in LIN mode only.<br><br>0 Sample point is RS9.<br>1 Sample point is RS13.                                                                                                                |
| 9<br>BESTP  | Bit Error Transmit Stop. This control bit defines the behavior of the eSCI Transmit Pin TXD while the bit error flag eSCI_IFSR1[BERR] is 1.<br><br><b>NOTE:</b> This bit is used in LIN mode only.<br><br>0 Application Data Values driven onto TXD pin.<br>1 Recessive Data Value 1 driven onto TXD pin.                                     |
| 10<br>RXPOL | RXD Pin polarity. This bit controls the polarity of the RXD pin. See <a href="#">Inverted Data Frame Formats</a> .<br><br>0 Normal Polarity.<br>1 Inverted Polarity.                                                                                                                                                                          |
| 11<br>PMSK  | Parity Bit Masking. This bit defines whether the received parity bit is presented in the related bit position in the <a href="#">SCI Data Register (eSCI_SDR)</a> .<br><br>0 The received parity bit is presented in the bit position related to the parity bit.<br>1 The value 0 is presented in the bit position related to the parity bit. |
| 12<br>ORIE  | Overrun Interrupt Enable. This bit controls the eSCI_IFSR1[OR] interrupt request generation.<br><br>0 OR interrupt request generation disabled.<br>1 OR interrupt request generation enabled.                                                                                                                                                 |
| 13<br>NFIE  | Noise Interrupt Enable. This bit controls the eSCI_IFSR1[NF] interrupt request generation.<br><br>0 NF interrupt request generation disabled.<br>1 NF interrupt request generation enabled.                                                                                                                                                   |
| 14<br>FEIE  | Frame Error Interrupt Enable. This bit controls the eSCI_IFSR1[FE] interrupt request generation.<br><br>0 FE interrupt request generation disabled.<br>1 FE interrupt request generation enabled.                                                                                                                                             |
| 15<br>PFIE  | Parity Error Interrupt Enable. This bit controls the eSCI_IFSR1[PF] interrupt request generation.                                                                                                                                                                                                                                             |

*Table continues on the next page...*



**eSCI\_CR2 field descriptions (continued)**

| Field | Description                               |
|-------|-------------------------------------------|
| 0     | PF interrupt request generation disabled. |
| 1     | PF interrupt request generation enabled.  |

**42.4.4 SCI Data Register (eSCI\_SDR)**

In SCI mode this register is used to provide transmit data and retrieve received data. In LIN mode any write access to this register is ignored. In case of read accesses to this register during data transmission/reception this register provides a part of the transmitted/received data.

If the application writes to the lower byte of this register (eSCI\_SDR[8:15 ]), the internal commit flag iCMT, which is not visible to the application, is set to indicate that the register has been updated and ready to transmit new data.

If the application reads from the lower byte of this register (eSCI\_SDR[8:15 ]), a signal is send to the internal receiver unit to indicate that the register was read and is ready to receive new data. The read access will not change the content of any register.

Address: 0h base + 6h offset = 6h

|       |    |    |     |   |      |   |   |   |   |   |    |    |    |    |    |    |
|-------|----|----|-----|---|------|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0  | 1  | 2   | 3 | 4    | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| Read  | RN | TN | ERR | 0 | RDTD |   |   |   |   |   |    |    |    |    |    |    |
| Write |    |    |     |   |      |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0  | 0  | 0   | 0 | 0    | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

**eSCI\_SDR field descriptions**

| Field   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>RN | Received Most Significant Bit. The semantic of this bit depends on the frame format selected by eSCI_CR3[M2], eSCI_CR1[M], and eSCI_CR1[PE].<br><br>[M2=0,M=1,PE=0]: value of received data bit 8 or address bit.<br><br>[M2=0,M=1,PE=1]: value of received parity bit if eSCI_CR2[PMSK]=0, 0 otherwise.<br><br>[M2=1,M=0,PE=1]: value of received parity bit if eSCI_CR2[PMSK]=0, 0 otherwise .<br><br>[M2=1,M=1,PE=1]: value of received parity bit if eSCI_CR2[PMSK]=0, 0 otherwise .<br><br>It is 0 for all other frame formats. |
| 1<br>TN | Transmit Most Significant Bit. The semantic of this bit depends on the frame format selected by eSCI_CR3[M2], eSCI_CR1[M], and eSCI_CR1[PE].<br><br>[M2=0,M=1,PE=0]: value to be transmitted as data bit 8 or address bit.<br><br>It is not used for all other frame formats.                                                                                                                                                                                                                                                        |

*Table continues on the next page...*

**eSCI\_SDR field descriptions (continued)**

| Field         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2<br>ERR      | <p>Receive Error Bit. This bit indicates the occurrence of the errors selected by the <a href="#">Control Register 3 (eSCI_CR3)</a> during the reception of the frame presented in <a href="#">SCI Data Register (eSCI_SDR)</a>. In case of an overrun error for subsequent frames this bit is set too.</p> <p>0 None of the selected errors occurred.<br/>1 At least one of the selected errors occurred.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 3<br>Reserved | <p>This field is reserved.<br/>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 4–15<br>RDTD  | <p>The contents of this field vary by bit and by semantic as follows.</p> <p><b>Most-significant 4 bits (RDTD[11:8]):</b></p> <p>Received Data. The semantic of this field depends on the frame format selected by eSCI_CR3[M2] and eSCI_CR1[M].</p> <p>[M2=1,M=1]: value of the received data bits 11:8. (Rx=BITx) .</p> <p>It is all 0 for all other frame formats .</p> <p><b>Fifthmost-significant bit (RDTD[7]):</b></p> <p>When read, this is Received Bit 7. The semantic of this bit depends on the format selected by eSCI_CR3[M2], eSCI_CR1[M], and eSCI_CR1[PE].</p> <p>[M2=0,M=0,PE=0]: value of received BIT7 or ADDR BIT.</p> <p>[M2=0;M=0,PE=1]: value of received PARITY BIT if eSCI_CR2[PMSK]=0, 0 otherwise.</p> <p>For all other frame formats it is the value of received BIT7.</p> <p>When written, this is Transmit Bit 7.The semantic of this bit depends on the format selected by eSCI_CR3[M2], eSCI_CR1[M], and eSCI_CR1[PE].</p> <p>[M2=0,M=0,PE=0]: value of transmit BIT7 or ADDR BIT.</p> <p>[M2=0;M=0,PE=1]: not used. PARITY BIT is generated internally before transmission.</p> <p>For all other frame formats it is the value of transmit BIT7.</p> <p><b>Remaining bits (RDTD[6:0]):</b></p> <p>When read, these are Received bits 6 to 0. Value of received BITx is shown in bit Rx.</p> <p>When written, these are Transmit bits 6 to 0. Value of bit Tx is transmitted in BITx.</p> |

## 42.4.5 Interrupt Flag and Status Register 1 (eSCI\_IFSR1)

This register provides interrupt flags that indicate the occurrence of module events. The related interrupt enable bits are located in [Control Register 1 \(eSCI\\_CR1\)](#) and [Control Register 2 \(eSCI\\_CR2\)](#).

Address: 0h base + 8h offset = 8h

| Bit   | 0    | 1   | 2    | 3    | 4    | 5    | 6    | 7    |
|-------|------|-----|------|------|------|------|------|------|
| Read  | TDRE | TC  | RDRF | IDLE | OR   | NF   | FE   | PF   |
| Write | w1c  | w1c | w1c  | w1c  | w1c  | w1c  | w1c  | w1c  |
| Reset | 1    | 0   | 0    | 0    | 0    | 0    | 0    | 0    |
| Bit   | 8    | 9   | 10   | 11   | 12   | 13   | 14   | 15   |
| Read  | 0    |     | DACT | BERR | WACT | LACT | TACT | RACT |
| Write |      |     |      | w1c  |      |      |      |      |
| Reset | 0    | 0   | 0    | 0    | 0    | 0    | 0    | 0    |

**eSCI\_IFSR1 field descriptions**

| Field     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>TDRE | Transmit Data Register Empty Interrupt Flag. This interrupt flag is set when the content of the <a href="#">SCI Data Register (eSCI_SDR)</a> was transferred into internal shift register.<br><b>NOTE:</b> This flag is set in SCI mode only.                                                                                                                                                                                                                                                                        |
| 1<br>TC   | Transmit Complete Interrupt Flag. This interrupt flag is set when a frame, break or idle character transmission has been completed and no data were written into <a href="#">SCI Data Register (eSCI_SDR)</a> after the last setting of the TDRE flag and the SBK bit in <a href="#">Control Register 1 (eSCI_CR1)</a> is 0.<br><b>NOTE:</b> This flag is set in SCI mode only. In case of no data available for transmission, it is also set if the preamble was transmitted after the enabling of the transmitter. |
| 2<br>RDRF | Receive Data Register Full Interrupt Flag. This interrupt flag is set when the payload data of a received frame was transferred into the <a href="#">SCI Data Register (eSCI_SDR)</a> and the receive DMA is disabled.<br><b>NOTE:</b> This flag is set in SCI mode only.                                                                                                                                                                                                                                            |
| 3<br>IDLE | Idle Line Interrupt Flag. This interrupt flag is set when an idle character was detected and the receiver is not in the wake-up state.<br><b>NOTE:</b> This flag is set in SCI mode only.                                                                                                                                                                                                                                                                                                                            |
| 4<br>OR   | Overrun Interrupt Flag. This interrupt flag is set when an overrun was detected as described in <a href="#">Receiver Overrun</a> .<br><b>NOTE:</b> This flag is set in SCI mode only.                                                                                                                                                                                                                                                                                                                                |
| 5<br>NF   | Noise Interrupt Flag. This interrupt flag is set when the receiver has detected noise during the reception of a frame, as described in <a href="#">Bit Sampling</a> .                                                                                                                                                                                                                                                                                                                                                |
| 6<br>FE   | Framing Error Interrupt Flag. This interrupt flag is set when the payload data of a received frame was transferred into the <a href="#">SCI Data Register (eSCI_SDR)</a> or <a href="#">LIN Receive Register (eSCI_LRR)</a> and the receiver has detected a framing error during the reception of that frame, as described in <a href="#">Stop Bit Verification</a> .                                                                                                                                                |

*Table continues on the next page...*

**eSCI\_IFSR1 field descriptions (continued)**

| Field           | Description                                                                                                                                                                                                                                                                                                                                                           |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 7<br>PF         | Parity Error Interrupt Flag. This interrupt flag is set when the payload data of a received frame was transferred into the <a href="#">SCI Data Register (eSCI_SDR)</a> and the receiver has detected a parity error for the character, as described in <a href="#">Reception Error Reporting</a> .<br><br><b>NOTE:</b> This flag is set in SCI mode only.            |
| 8–9<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                               |
| 10<br>DACT      | DMA Active. The status bit is set when a transmit or receive DMA request is pending .<br><br>0 No DMA request pending.<br>1 DMA request pending.                                                                                                                                                                                                                      |
| 11<br>BERR      | Bit Error Interrupt Flag. This flag is set when a bit error was detected as described in <a href="#">Standard Bit Error Detection</a> .<br><br>Note: This flag is set in LIN mode only.                                                                                                                                                                               |
| 12<br>WACT      | LIN Wake-Up Active. The status bit is set as long as the LIN wakeup engine receives a LIN wake-up signal .<br><br>0 No LIN wakeup signal reception in progress .<br>1 LIN wakeup signal reception in progress .                                                                                                                                                       |
| 13<br>LACT      | LIN Active. This status bit is set as long as the LIN protocol engine is about to transmit or receive LIN frames .<br><br>0 No LIN frame transmission or reception in progress .<br>1 LIN frame transmission or reception in progress .                                                                                                                               |
| 14<br>TACT      | Transmitter Active. This status bit is set as long as the transmission of a frame or special character is ongoing .<br><br>0 No transmission in progress .<br>1 Transmission in progress .                                                                                                                                                                            |
| 15<br>RACT      | Receiver Active. This status bit is set as long as the receive is active. The set and clear conditions for the SCI mode are described in <a href="#">Receiver States and Transitions</a> . The set and clear conditions for the LIN mode are described in <a href="#">LIN byte field reception</a> .<br><br>0 No reception in progress .<br>1 Reception in progress . |

## 42.4.6 Interrupt Flag and Status Register 2 (eSCI\_IFSR2)

This register provides interrupt flags that indicate the occurrence of LIN related events. The related interrupt enable bits are located in [LIN Control Register 1 \(eSCI\\_LCR1\)](#) and [LIN Control Register 2 \(eSCI\\_LCR2\)](#). All interrupt flags in this register will be set in LIN mode only .

Address: 0h base + Ah offset = Ah

|       |       |       |       |     |       |      |       |      |
|-------|-------|-------|-------|-----|-------|------|-------|------|
| Bit   | 0     | 1     | 2     | 3   | 4     | 5    | 6     | 7    |
| Read  | RXRDY | TXRDY | LWAKE | STO | PBERR | CERR | CKERR | FRC  |
| Write | w1c   | w1c   | w1c   | w1c | w1c   | w1c  | w1c   | w1c  |
| Reset | 0     | 1     | 0     | 0   | 0     | 0    | 0     | 0    |
| Bit   | 8     | 9     | 10    | 11  | 12    | 13   | 14    | 15   |
| Read  | 0     |       |       |     |       |      | UREQ  | OVFL |
| Write |       |       |       |     |       |      | w1c   | w1c  |
| Reset | 0     | 0     | 0     | 0   | 0     | 0    | 0     | 0    |

**eSCI\_IFSR2 field descriptions**

| Field            | Description                                                                                                                                                                                                                                                                                                                                       |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>RXRDY       | Receive Data Ready Interrupt Flag. This interrupt flag is set when the payload data of a received frame was transferred into the <a href="#">LIN Receive Register (eSCI_LRR)</a> and the receive DMA is disabled.<br><b>NOTE:</b> This bit should be cleared only after reading the data from the <a href="#">LIN Receive Register (eSCI_LRR)</a> |
| 1<br>TXRDY       | Transmit Data Ready Interrupt Flag. This interrupt flag is set when<br>a) the content of the <a href="#">LIN Transmit Register (eSCI_LTR)</a> was processed by the LIN PE to generate frame header or frame transmit data, or<br>b) when the module has transmitted a LIN wakeup signal frame.                                                    |
| 2<br>LWAKE       | LIN Wake-up Received Interrupt Flag. This interrupt flag is set when a LIN Wake-up character was received, as described in <a href="#">LIN Wake Up</a> .                                                                                                                                                                                          |
| 3<br>STO         | Slave Timeout Interrupt Flag. This interrupt flag is set when a Slave-Not-Responding-Error is detected. A detailed description is given in <a href="#">Slave-Not-Responding-Error Detection</a> .                                                                                                                                                 |
| 4<br>PBERR       | Physical Bus Error Interrupt Flag. This interrupt flag is set when the receiver input remains unchanged for at least 31 RCLK clock cycles after the start of a byte transmission, as described in <a href="#">LIN Error Reporting</a> .                                                                                                           |
| 5<br>CERR        | CRC Error Interrupt Flag. This interrupt flag is set when an incorrect CRC pattern was detected for a received LIN frame.                                                                                                                                                                                                                         |
| 6<br>CKERR       | Checksum Error Interrupt Flag. This interrupt flag is set when a checksum error was detected for a received LIN frame.                                                                                                                                                                                                                            |
| 7<br>FRC         | Frame Complete Interrupt Flag. This interrupt flag is set when a LIN TX frame has been completely transmitted or a LIN RX frame has been completely received.                                                                                                                                                                                     |
| 8–13<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                           |

*Table continues on the next page...*

**eSCI\_IFSR2 field descriptions (continued)**

| Field      | Description                                                                                                                                                                            |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 14<br>UREQ | Unrequested Data Received Interrupt Flag. This interrupt flag is set when unrequested activity has been detected on the LIN bus, as described in <a href="#">LIN Error Reporting</a> . |
| 15<br>OVFL | Overflow Interrupt Flag. This interrupt flag is set when an overflow as described in <a href="#">Overflow Detection</a> was detected.                                                  |

**42.4.7 LIN Control Register 1 (eSCI\_LCR1)**

This register provides control bits to control and configure the LIN hardware. This register provides the interrupt enable bits for the interrupt flags in [Interrupt Flag and Status Register 2 \(eSCI\\_IFSR2\)](#).

Address: 0h base + Ch offset = Ch

| Bit   | 0    | 1  | 2   | 3 | 4 | 5 | 6    | 7   | 8    | 9    | 10   | 11   | 12   | 13  | 14   | 15   |
|-------|------|----|-----|---|---|---|------|-----|------|------|------|------|------|-----|------|------|
| Read  | LRES | 0  | WUD |   | 0 |   | PRTY | LIN | RXIE | TXIE | WUIE | STIE | PBIE | CIE | CKIE | FCIE |
| Write |      | WU |     |   |   |   |      |     |      |      |      |      |      |     |      |      |
| Reset | 0    | 0  | 0   | 0 | 0 | 0 | 0    | 0   | 0    | 0    | 0    | 0    | 0    | 0   | 0    | 0    |

**eSCI\_LCR1 field descriptions**

| Field           | Description                                                                                                                                                                                                                                                                      |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>LRES       | LIN Protocol Engine Stop and Reset. This bit is used to stop and reset the LIN protocol engine as described in <a href="#">LIN Protocol Engine Stop and Reset</a> .<br><br>0 LIN protocol engine is operational.<br>1 LIN protocol engine is reset and stopped.                  |
| 1<br>WU         | LIN Bus Wake-Up Trigger. This bit is used to trigger the generation of a wake-up signal frame on the LIN bus, as described in <a href="#">LIN Wake Up</a> .<br><br>0 Write has no effect.<br>1 Write triggers the generation of a wake-up signal.                                |
| 2–3<br>WUD      | LIN Bus Wake-Up Delimiter Time. This field determines how long the LIN protocol engine waits after the end of the transmitted wake-up signal, before starting the next LIN frame transmission.<br><br>00 3 bit times.<br>01 7 bit times.<br>10 31 bit times.<br>11 63 bit times. |
| 4–5<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                          |
| 6<br>PRTY       | Parity Generation Control. This bit controls the generation of the two parity bits in the LIN header.                                                                                                                                                                            |

Table continues on the next page...

**eSCI\_LCR1 field descriptions (continued)**

| Field      | Description                                                                                                                                                                                                         |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|            | 0 Parity bits generation disabled.<br>1 Parity bits generation enabled.                                                                                                                                             |
| 7<br>LIN   | LIN Mode Control. This bit controls whether the device is in SCI or LIN Mode.<br><br>0 SCI Mode.<br>1 LIN Mode.                                                                                                     |
| 8<br>RXIE  | Receive Data Ready Interrupt Enable. This bit controls the eSCI_IFSR2[RXRDY] interrupt request generation.<br><br>0 RXRDY interrupt request generation disabled.<br>1 RXRDY interrupt request generation enabled.   |
| 9<br>TXIE  | Transmit Data Ready Interrupt Enable. This bit controls the eSCI_IFSR2[TXRDY] interrupt request generation.<br><br>0 TXRDY interrupt request generation disabled.<br>1 TXRDY interrupt request generation enabled.  |
| 10<br>WUIE | LIN Wake-up Received Interrupt Enable. This bit controls the eSCI_IFSR2[LWAKE] interrupt request generation.<br><br>0 LWAKE interrupt request generation disabled.<br>1 LWAKE interrupt request generation enabled. |
| 11<br>STIE | Slave Timeout Flag Interrupt Enable. This bit controls the eSCI_IFSR2[STO] interrupt request generation.<br><br>0 STO interrupt request generation disabled.<br>1 STO interrupt request generation enabled.         |
| 12<br>PBIE | Physical Bus Error Interrupt Enable. This bit controls the eSCI_IFSR2[PBERR] interrupt request generation.<br><br>0 PBERR interrupt request generation disabled.<br>1 PBERR interrupt request generation enabled.   |
| 13<br>CIE  | CRC Error Interrupt Enable. This bit controls the eSCI_IFSR2[CERR] interrupt request generation.<br><br>0 CERR interrupt request generation disabled.<br>1 CERR interrupt request generation enabled.               |
| 14<br>CKIE | Checksum Error Interrupt Enable. This bit controls the eSCI_IFSR2[CKERR] interrupt request generation.<br><br>0 CKERR interrupt request generation disabled.<br>1 CKERR interrupt request generation enabled.       |
| 15<br>FCIE | Frame Complete Interrupt Enable. This bit controls the eSCI_IFSR2[FRC] interrupt request generation.<br><br>0 FRC interrupt request generation disabled.<br>1 FRC interrupt request generation enabled.             |

### 42.4.8 LIN Control Register 2 (eSCI\_LCR2)

This register provides the interrupt enable bits for the interrupt flags in [Interrupt Flag and Status Register 2 \(eSCI\\_IFSR2\)](#).

Address: 0h base + Eh offset = Eh

|       |   |   |   |   |   |   |      |      |   |   |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|------|------|---|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6    | 7    | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| Read  | 0 |   |   |   |   |   | UQIE | OFIE | 0 |   |    |    |    |    |    |    |
| Write |   |   |   |   |   |   |      |      |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0    | 0    | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

eSCI\_LCR2 field descriptions

| Field            | Description                                                                                                                                                                                                       |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–5<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                           |
| 6<br>UQIE        | Unrequested Data Received Interrupt Enable. This bit controls the eSCI_IFSR2[UREQ] interrupt request generation.<br>0 UREQ interrupt request generation disabled.<br>1 UREQ interrupt request generation enabled. |
| 7<br>OFIE        | Overflow Interrupt Enable. This bit controls the eSCI_IFSR2[OVFL] interrupt request generation.<br>0 OVFL interrupt request generation disabled.<br>1 OVFL interrupt request generation enabled.                  |
| 8–15<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                           |

### 42.4.9 LIN Transmit Register (eSCI\_LTR)

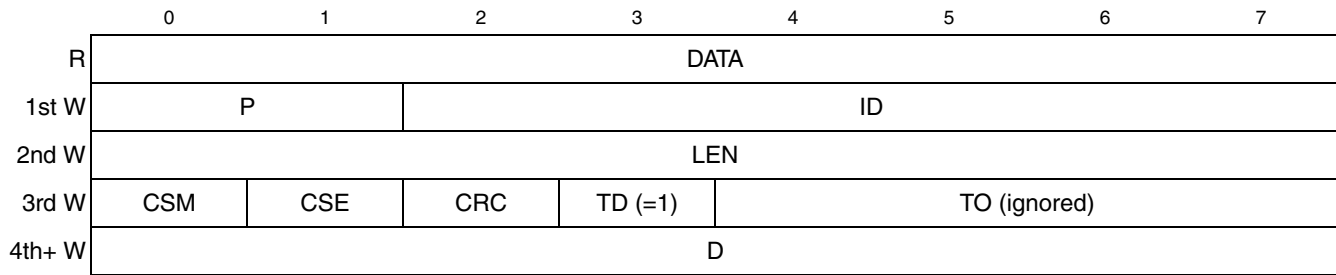
This register is used by the application to initiate the LIN frame header generation for both LIN TX frames and LIN RX frames. If a LIN TX frame is generated, this register is used to provide the payload data for the LIN TX frame.

If the LIN PE is in the idle state (eSCI\_LCR1[LRES] = 1) or performs a wakeup, each write access to this register is ignored.

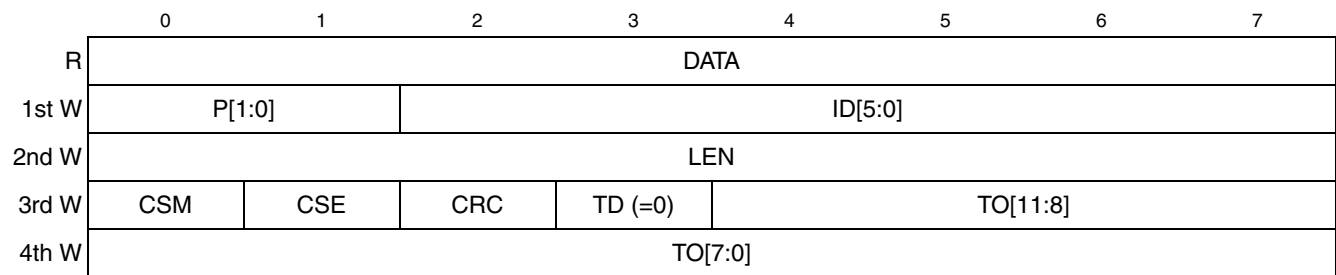
In case of a read access, the register provides the last data written into this register in the DATA field.

If the application initiates a LIN TX frame transfer, i.e the TD bit is set to 1, the content and usage shown in [Figure 42-11](#) applies. The initiation and transmit of a TX frame is described in [LIN TX Frame generation](#).



**Figure 42-11. LIN Transmit Register (eSCI\_LTR) - LIN TX frame generation**

If the application initiates an LIN RX frame, i.e the TD bit is set to 0, the content and usage shown in [Figure 42-12](#) applies. The initiation and transmit of a RX frame is described in [LIN RX frame generation](#).

**Figure 42-12. LIN Transmit Register (eSCI\_LTR) - LIN RX frame generation**

Each successful write access to this register increments the internal write access counter and enables the writing to the next field. The write access counter is reset if

- the LIN PE is in the idle state (eSCI\_LCR1[LRES] = 1)
- a LIN TX frame was completely transmitted (eSCI\_IFSR1[FRC] was set to 1)
- a LIN RX frame was completely received (eSCI\_IFSR1[FRC] was set to 1)

**Table 42-15. Structure of the LTR[DATA] when written to**

| Field | Description                                                                                                                                                                                                                                           |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| P     | Identifier Parity. This field provides the identifier parity which is used to create the protected identifier if the automatic identifier parity generation is disabled, i.e the PRTY bit in <a href="#">LIN Control Register 1 (eSCI_LCR1)</a> is 0. |
| ID    | Identifier. This field is used for the identifier field in the protected identifier.                                                                                                                                                                  |
| LEN   | Frame Length. This field defines the number of data bytes to be transmitted or received.                                                                                                                                                              |
| CSM   | Checksum Model. This bit controls the checksum calculation model used.<br>0: Classic Checksum Model (LIN 1.3).<br>1: Enhanced Checksum Model (LIN 2.0).                                                                                               |
| CSE   | Checksum Enable. This bit control the generation and checking of the checksum byte.<br>0: No generation and checking of checksum byte.<br>1: Generation and checking of checksum byte.                                                                |
| CRC   | CRC Enable. This bit controls the generation of checking standard or enhanced LIN frames, which are described in <a href="#">LIN frame formats</a> .                                                                                                  |

Table continues on the next page...

**Table 42-15. Structure of the LTR[DATA] when written to (continued)**

| Field | Description                                                                                                                                                                                                                                                                                                                       |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | 0: Standard LIN frame generation and checking.<br>1: Enhanced LIN frame generation and checking.                                                                                                                                                                                                                                  |
| TD    | Transfer Direction. This bit controls the transfer direction of the data, CRC, and checksum byte fields.<br>0: Data, CRC, and Checksum byte fields received, described in <a href="#">LIN RX frame generation</a> .<br>1: Data, CRC, and Checksum byte fields transmitted, described in <a href="#">LIN TX Frame generation</a> . |
| TO    | Timeout Value. The content of the field depends on the transfer direction.<br>RX frame: Defines the time available for a complete RX frame transfer, as described in the note in <a href="#">Fast Bit Error Detection</a> .<br>TX frame: Must be set to 0.                                                                        |
| D     | Transmit Data . Data bits for transmission.                                                                                                                                                                                                                                                                                       |

Address: 0h base + 10h offset = 10h

|       |      |   |   |   |   |   |   |   |
|-------|------|---|---|---|---|---|---|---|
| Bit   | 0    | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Read  | DATA |   |   |   |   |   |   |   |
| Write |      |   |   |   |   |   |   |   |
| Reset | 0    | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**eSCI\_LTR field descriptions**

| Field       | Description                                                                                                                  |
|-------------|------------------------------------------------------------------------------------------------------------------------------|
| 0–7<br>DATA | When read: The value written in the most recent successful write access<br>When written: See the description in this section |

**42.4.10 LIN Receive Register (eSCI\_LRR)**

This register provides the data bytes of received in case of an LIN RX frame was initiated.

Address: 0h base + 14h offset = 14h

|       |   |   |   |   |   |   |   |   |
|-------|---|---|---|---|---|---|---|---|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Read  | D |   |   |   |   |   |   |   |
| Write |   |   |   |   |   |   |   |   |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**eSCI\_LRR field descriptions**

| Field    | Description                                                                  |
|----------|------------------------------------------------------------------------------|
| 0–7<br>D | Receive Data . This field provides the data bytes of received LIN RX frames. |

### 42.4.11 LIN CRC Polynomial Register (eSCI\_LPR)

This register provides the CRC polynomial for generation and processing of CRC-enhanced LIN frames.

Address: 0h base + 18h offset = 18h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| Read  | P |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Write |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0  | 1  | 1  | 0  | 0  | 1  |

**eSCI\_LPR field descriptions**

| Field     | Description                                                                                                                                                                                          |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>P | Polynomial bit $x^{P[n]}$ . Used to define the LIN polynomial.<br>Reset value results in $x^{15} + x^{14} + x^{10} + x^8 + x^7 + x^4 + x^3 + 1$ , which is the polynomial used for the CAN protocol. |

### 42.4.12 Control Register 3 (eSCI\_CR3)

This register is used to control the frame formats and the generation of the ERR bit in the [SCI Data Register \(eSCI\\_SDR\)](#).

Address: 0h base + 1Ah offset = 1Ah

|       |   |   |   |   |     |      |      |    |   |   |    |    |    |    |    |    |
|-------|---|---|---|---|-----|------|------|----|---|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4   | 5    | 6    | 7  | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| Read  | 0 |   |   |   | ERO | ERFE | ERPE | M2 | 0 |   |    |    |    |    |    |    |
| Write |   |   |   |   | E   |      |      |    |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0   | 0    | 0    | 0  | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

**eSCI\_CR3 field descriptions**

| Field           | Description                                                                                                                                                                            |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–3<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                |
| 4<br>EROE       | ERR flag overrun enable.<br>0 eSCI_SDR[ERR] flag not affected by overrun detection.<br>1 eSCI_SDR[ERR] flag is set on overrun detection during frame reception.                        |
| 5<br>ERFE       | ERR flag frame error enable.<br>0 eSCI_SDR[ERR] flag not affected by frame error detection.<br>1 eSCI_SDR[ERR] flag is set on frame error detection for the data provided in eSCI_SDR. |

*Table continues on the next page...*

**eSCI\_CR3 field descriptions (continued)**

| Field            | Description                                                                                                                                                                                                                                            |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 6<br>ERPE        | ERR flag parity error enable.<br><br>0 eSCI_SDR[ERR] flag not affected by parity error detection.<br>1 eSCI_SDR[ERR] flag is set on parity error detection for the data provided in eSCI_SDR.                                                          |
| 7<br>M2          | Frame Format Mode 2. This control bit together with the M bit of the <a href="#">Control Register 1 (eSCI_CR1)</a> controls the frame format used. The supported frame formats and the related settings are defined in <a href="#">Frame Formats</a> . |
| 8–15<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                |

## 42.5 Functional Description

This section provides a complete functional description of the eSCI block, detailing the operation of the design from the end user perspective in a number of subsections.

### 42.5.1 Module Control

The operational mode of the module is controlled by the MDIS bit in the Control Register 2 (eSCI\_CR2). The module can transmit and receives data when it is enabled, i.e MDIS=0.

### 42.5.2 Frame Formats

The eSCI module uses the standard NRZ mark/space data format. The eSCI supports three basic frame types, which are the data frames, break characters, and idle characters.

#### 42.5.2.1 Data Frame Formats

Each data frame contains a character that is surrounded by a start bit, an optional parity or address bit, and one or two stop bits. The supported data frame formats for transmission and reception are specified in [Table 42-19](#). The supported data frame formats for reception only are specified in [Table 42-20](#).

**Table 42-19. Supported Data Frame Formats for RX and TX**

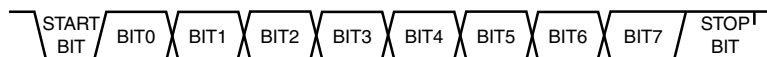
| Control                                    |          |    |      | Frame Content |                |                           |             |           |
|--------------------------------------------|----------|----|------|---------------|----------------|---------------------------|-------------|-----------|
| eSCI_CR3                                   | eSCI_CR1 |    |      | Start Bits    | Payload Bits   |                           |             | Stop Bits |
| M2                                         | M        | PE | WAKE |               | Character Bits | Address Bits <sup>1</sup> | Parity Bits |           |
| LIN byte fields (Figure 42-16)             |          |    |      |               |                |                           |             |           |
| 0                                          | 0        | 0  | 0    | 1             | 8              | 0                         | 0           | 1         |
| SCI Frames (8 payload bits)(Figure 42-17)  |          |    |      |               |                |                           |             |           |
| 0                                          | 0        | 0  | 0    | 1             | 8              | 0                         | 0           | 1         |
| 0                                          | 0        | 0  | 1    | 1             | 7              | 1                         | 0           | 1         |
| 0                                          | 0        | 1  | 0    | 1             | 7              | 0                         | 1           | 1         |
| SCI Frames (9 payload bits) (Figure 42-18) |          |    |      |               |                |                           |             |           |
| 0                                          | 1        | 0  | 0    | 1             | 9              | 0                         | 0           | 1         |
| 0                                          | 1        | 0  | 1    | 1             | 8              | 1                         | 0           | 1         |
| 0                                          | 1        | 1  | 0    | 1             | 8              | 0                         | 1           | 1         |

1. The address bit identifies the frame as an address character. See [Multiprocessor Communication](#).

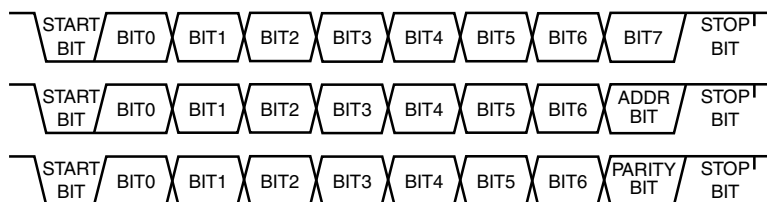
**Table 42-20. Supported Data Frame Formats for RX only**

| Control                                                      |          |    |      | Frame Content |                |              |             |           |
|--------------------------------------------------------------|----------|----|------|---------------|----------------|--------------|-------------|-----------|
| eSCI_CR3                                                     | eSCI_CR1 |    |      | Start Bits    | Payload Bits   |              |             | Stop Bits |
| M2                                                           | M        | PE | WAKE |               | Character Bits | Address Bits | Parity Bits |           |
| SCI Frames (2 stop bits) (see <a href="#">Figure 42-19</a> ) |          |    |      |               |                |              |             |           |
| 1                                                            | 0        | 1  | 0    | 1             | 8              | 0            | 1           | 2         |
| 1                                                            | 1        | 1  | 0    | 1             | 12             | 0            | 1           | 2         |

The structure of the LIN byte fields in normal polarity is shown in [Figure 42-16](#).

**Figure 42-16. LIN Byte Field Format**

The structures of the supported SCI frame formats with 8 payload bits are shown in [Figure 42-17](#).

**Figure 42-17. SCI Frame Formats (8 payload bits)**

The structures of the supported SCI frame formats with 9 payload bits are shown in [Figure 42-18](#).

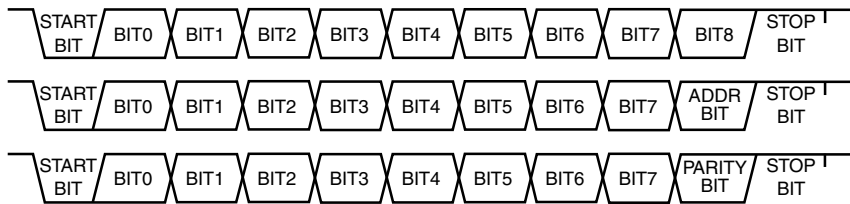


**Figure 42-18. SCI Frame Formats (9 payload bits)**

The structures of the supported SCI frame formats with 2 stop bits in normal polarity are shown in [Figure 42-19](#). This frame format is supported for reception only.

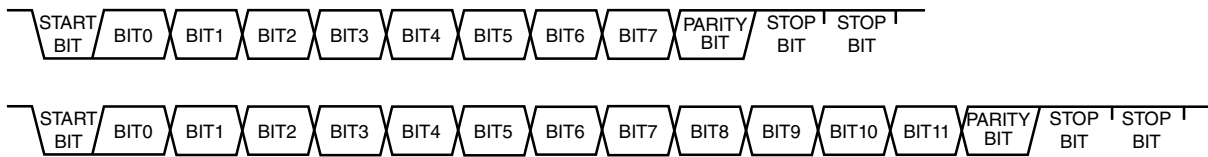


**Figure 42-19. SCI Frame Formats (2 stop bits)**

**42.5.2.1.1 Inverted Data Frame Formats**

The structures of the supported data frame formats in inverted polarity are shown in the following figure. These frame types are supported for reception only. The polarity of the RXD pin is controlled by the RXPOL bit in the Control Register 2 (eSCI\_CR2).

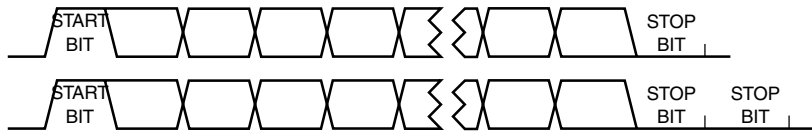


**Figure 42-20. Inverted SCI Frame Formats**

**42.5.2.2 Break Character Formats**

The supported break character formats are specified in [Table 42-21](#).

**Table 42-21. Supported Break Character Formats**

| Control <sup>1</sup>                                 |          |          | Break Character Content |                |              |
|------------------------------------------------------|----------|----------|-------------------------|----------------|--------------|
| eSCI_CR3                                             | eSCI_CR1 | eSCI_CR2 | Start Bit               | Character Bits | Delemit Bits |
| M2                                                   | M        | BRCL     |                         |                |              |
| LIN Break Symbol (see <a href="#">Figure 42-21</a> ) |          |          |                         |                |              |
| 0                                                    | 0        | 0        | 1                       | 9              | 1            |

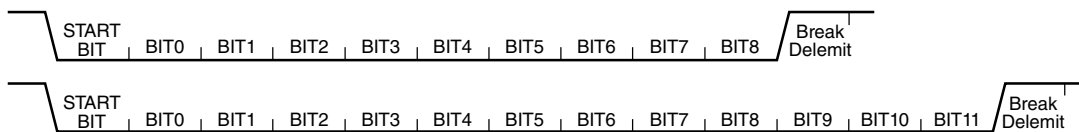
Table continues on the next page...

**Table 42-21. Supported Break Character Formats (continued)**

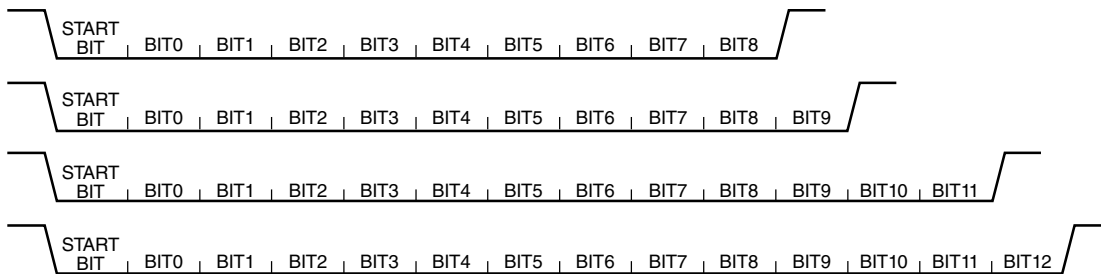
| Control <sup>1</sup>                                    |          |          | Break Character Content |                |              |
|---------------------------------------------------------|----------|----------|-------------------------|----------------|--------------|
| eSCI_CR3                                                | eSCI_CR1 | eSCI_CR2 | Start Bit               | Character Bits | Delemit Bits |
| M2                                                      | M        | BRCL     |                         |                |              |
| 0                                                       | 0        | 1        | 1                       | 12             | 1            |
| SCI Break Character (see <a href="#">Figure 42-22</a> ) |          |          |                         |                |              |
| 0                                                       | 0        | 0        | 1                       | 9              | 0            |
| 0                                                       | 0        | 1        | 1                       | 12             | 0            |
| 0                                                       | 1        | 0        | 1                       | 10             | 0            |
| 0                                                       | 1        | 1        | 1                       | 13             | 0            |

1. All codings which are not listed are reserved and must not be used.

The structure and content of the LIN break symbols is shown in [Figure 42-21](#).

**Figure 42-21. LIN Break Symbol Format**

The structure and content of the SCI break characters is shown in [Figure 42-22](#).

**Figure 42-22. SCI Break Character Formats**

### 42.5.2.3 Idle Character Formats

An idle character is a sequence of bits with the value 1. The supported idle character formats are specified in [Table 42-22](#). The preamble has the same structure and content as an idle character.

**Table 42-22. Supported Idle Character Formats**

| Control                                             |             | Idle Character Length |
|-----------------------------------------------------|-------------|-----------------------|
| eSCI_CR3[M2]                                        | eSCI_CR1[M] |                       |
| Idle Characters (see <a href="#">Figure 42-23</a> ) |             |                       |

Table continues on the next page...

Table 42-22. Supported Idle Character Formats (continued)

| Control      |             | Idle Character Length |
|--------------|-------------|-----------------------|
| eSCI_CR3[M2] | eSCI_CR1[M] |                       |
| 0            | 0           | 10                    |
| 0            | 1           | 11                    |
| 1            | 0           | 12                    |
| 1            | 1           | 16                    |

The structure and content of the idle characters is shown in [Figure 42-23](#).

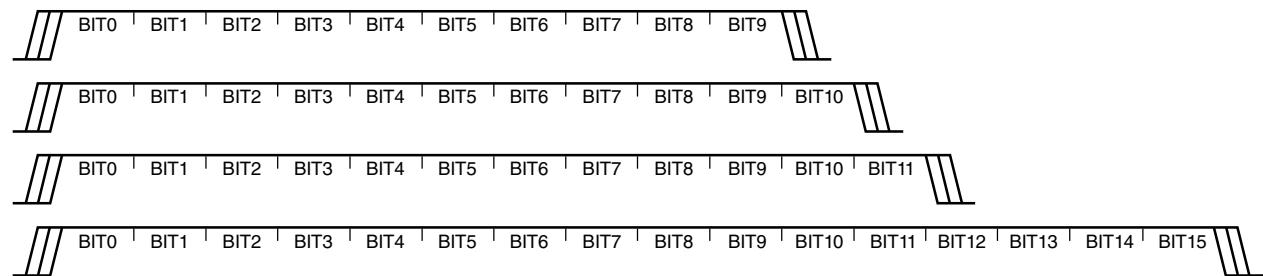


Figure 42-23. Idle Character Formats

42.5.3 Baud Rate and Clock Generation

A 13-bit modulus counter in the baud rate generator derives the baud rate for both the receiver and the transmitter. The value written to the SBR field in the Baud Rate Register (eSCI\_BRR) determines the module clock divisor. The baud rate clock is synchronized with the bus clock and drives the receiver. The baud rate clock divided by 16 drives the transmitter. The receiver has an acquisition rate of 16 samples per bit time.

The baud rate generator is enabled when the TE bit or RE bit in the Control Register 1 (eSCI\_CR1) is set to 1 for the first time. The baud rate generator is disabled when SBR = 0.

Baud rate generation is subject to one source of error:

- Integer division of the module clock may not give the exact required target baud rate.

The following table lists some examples of achieving target baud rates with a module clock frequency of MCLK = 10.2 MHz.

Table 42-23. Baud Rates Error Example (MCLK = 10.2 MHz)

| eSCI_BRR[SBR] | RCLK (Hz) | TCLK (Hz) | Target Baud Rate | Error (%) |
|---------------|-----------|-----------|------------------|-----------|
| 17            | 600,000.0 | 37,500.0  | 38,400           | 2.3       |

Table continues on the next page...



**Table 42-23. Baud Rates Error Example (MCLK = 10.2 MHz) (continued)**

| eSCI_BRR[SBR] | RCLK (Hz) | TCLK (Hz) | Target Baud Rate | Error (%) |
|---------------|-----------|-----------|------------------|-----------|
| 33            | 309,090.9 | 19,318.2  | 19,200           | .62       |
| 66            | 154,545.5 | 9659.1    | 9600             | .62       |
| 133           | 76,691.7  | 4793.2    | 4800             | .14       |
| 266           | 38,345.9  | 2396.6    | 2400             | .14       |
| 531           | 19,209.0  | 1200.6    | 1200             | .11       |
| 1062          | 9604.5    | 600.3     | 600              | .05       |
| 2125          | 4800.0    | 300.0     | 300              | .00       |
| 4250          | 2400.0    | 150.0     | 150              | .00       |
| 5795          | 1760.1    | 110.0     | 110              | .00       |

### 42.5.3.1 Module Clock

For the frequency of module clock MCLK and the system bus clock, see the chip-specific eSCI information. MCLK has the same phase as the system bus clock.

### 42.5.3.2 Transmitter Clock

The transmitter clock TCLK is used to drive the data to the serial bus via the TXD pin. It is derived from the system bus clock by the baud rate generator. The baud rate generator is controlled by the value of the SBR field in the Baud Rate Register (eSCI\_BRR). The frequency of the transmitter clock is determined by the following equation and defines the length of the transmitted bits, which is denoted as the *bit time*.

$$f_{\text{TCLK}} = \frac{f_{\text{MCLK}}}{16 \cdot \text{SBR}}$$

### 42.5.3.3 Receiver Clock

The receiver clock RCLK is used to sample the data received on the RXD or TXD pin. It is derived from the system bus clock by the baud rate generator. The baud rate generator is controlled by the value of the SBR field in the Baud Rate Register (eSCI\_BRR). The frequency of the receiver sample clock is determined by the following equation.

$$f_{\text{RT}} = \frac{f_{\text{MCLK}}}{\text{SBR}}$$

The frequency of the receiver clock is 16 times the frequency of the transmitter clock, thus each bit is sampled with 16 samples. Each of the 16 samples of a bit has a sample number assigned, which is defined by the receiver sample counter RSC. The n-th sample is denoted by RS<sub>n</sub>. The receiver sample counter RSC is updated with each rising edge of the receiver clock RCLK.

## 42.5.4 Baud Rate Tolerance

A transmitting device may be operating at a baud rate below or above the receiver baud rate. Accumulated bit time misalignment can cause one of the three stop bit data samples RS<sub>8</sub>, RS<sub>9</sub>, and RS<sub>10</sub> to fall outside the actual stop bit. A noise error will occur if the stop bit sample RS<sub>8</sub>, RS<sub>9</sub>, and RS<sub>10</sub> samples are not all the same logical value 1. A framing error will occur if the receiver clock is misaligned in such a way that the majority of the RS<sub>8</sub>, RS<sub>9</sub>, and RS<sub>10</sub> stop bit samples are a logic zero.

### 42.5.4.1 Faster Receiver Tolerance

In this case the receiver has a higher baud rate than the transmitter, thus the stop bit sampling starts already in the last transmitted payload bit. To ensure the correct, noise and framing error free reception of the stop bit, the samples RS<sub>8</sub>, RS<sub>9</sub>, and RS<sub>10</sub> must be located in the transmitted stop bit as shown in the following figure.

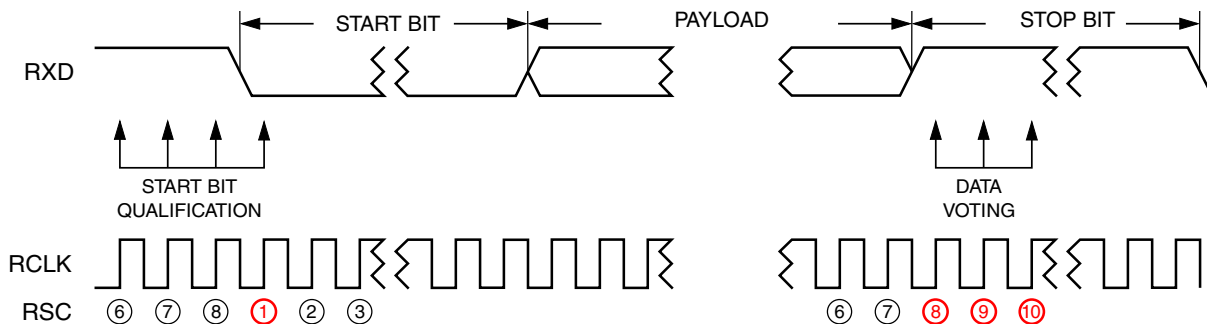


**Figure 42-24. Faster Receiver**

The maximum tolerance that ensures error free reception can be calculated with the assumption that RS<sub>7</sub> is sampled during the last transmitted payload bit and RS<sub>8</sub> is sampled in the stop bit.

For an frame with  $n$  payload bits the transmitter starts the transmission of the stop bit

$$t_{XSTOP} = (n + 1) \cdot 16 \cdot RT_{TR}$$

after the start of the transmission of the start bit.

For an frame with  $n$  payload bits the receiver samples the RS8 sample of the stop bit

$$rx_{STOP} = (n + 1) \cdot 16 \cdot RT_{RE} + 7 \cdot RT_{RE}$$

after the successful qualification of the start bit.

To ensure error free reception of the stop bit, the transmitter must start the transmission of the stop bit before the receiver samples RSC8.

$$tx_{STOP} < rx_{STOP}$$

The maximum percent difference between the receiver baud rate and the transmitter baud rate is:

$$\Delta \text{baudrate} \leq \left( \frac{rx_{STOP} - tx_{STOP}}{tx_{STOP}} \right) \times 100$$

The maximum percent differences for the supported frames is given in the following table.

**Table 42-24. Faster Receiver Maximum Tolerance**

| payload bits | max baudrate difference | tx <sub>STOP</sub> | rx <sub>STOP</sub> |
|--------------|-------------------------|--------------------|--------------------|
| 8            | 4.63%                   | 144                | 151                |
| 9            | 4.19%                   | 160                | 167                |
| 13           | 3.03%                   | 224                | 231                |

#### 42.5.4.2 Slower Receiver Tolerance

In this case the receiver has a slower baud rate than the transmitter, thus the stop bit sampling is still running while the next start bit is already transmitted. To ensure the correct, noise and framing error free reception of the stop bit, the samples RS8, RS9, and RS10 must be located in the transmitted stop bit as shown in the following figure.

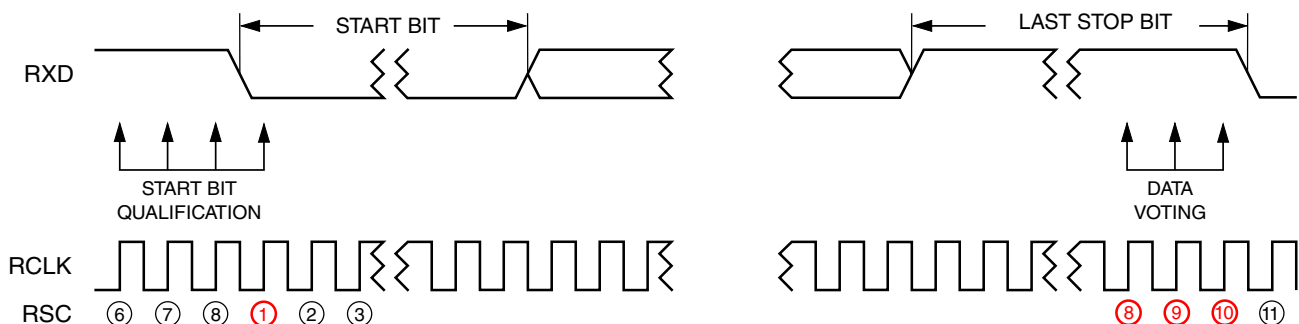


**Figure 42-25. Slower Receiver**

## Functional Description

The maximum tolerance that ensures error free reception can be calculated with the assumption that RS11 is sampled in the transmitted start bit and RS10 is sampled in the last stop bit.

For a frame with  $n$  payload bits and  $s$  stop bits, the transmitter starts the transmission of the next start bit

$$t_{xSTART} = (n + s + 1) \cdot 16 \cdot RT_{TR}$$

after the start of the transmission of the previous start bit.

For a frame with  $n$  payload bits and  $s$  stop bits, the receiver samples the RS10 sample of the last stop bit

$$r_{xSTOP} = (n + s) \cdot 16 \cdot RT_{RE} + 9 \cdot RT_{RE}$$

after the successful qualification of the start bit.

To ensure error free reception of the last stop bit, the transmitter must start the transmission of the start bit after the receiver samples RS10.

$$r_{xSTOP} < t_{xSTART}$$

The maximum percent difference between the receiver baud rate and the transmitter baud rate is:

$$\Delta \text{baudrate} \leq \left( \frac{t_{xSTART} - r_{xSTOP}}{t_{xSTART}} \right) \times 100$$

The maximum percent differences for the supported frames is given in the following table.

**Table 42-25. Slower Receiver Maximum Tolerance**

| payload bits | stop bits | max baudrate difference | $r_{xSTOP}$ | $t_{xSTART}$ |
|--------------|-----------|-------------------------|-------------|--------------|
| 8            | 1         | 4.37%                   | 153         | 160          |
| 9            | 1         | 3.97%                   | 169         | 176          |
| 9            | 2         | 3.57%                   | 185         | 196          |
| 13           | 2         | 2.73%                   | 249         | 256          |

## 42.5.5 SCI Mode

### 42.5.5.1 SCI Mode Configuration

The application must configure the following bits and fields in order to achieve correct SCI operation.

- enable *SCI* Mode
  - LIN Control Register 1 (eSCI\_LCR1)[LIN]:= 0
- select *baud rate*
  - Baud Rate Register (eSCI\_BRR)[SBR]
- select *receiver input* mode
  - Control Register 1 (eSCI\_CR1)[LOOPS]
  - Control Register 1 (eSCI\_CR1)[RSRC]
- select *frame format*
  - Control Register 1 (eSCI\_CR1)[M]
  - Control Register 1 (eSCI\_CR1)[PE]
  - Control Register 1 (eSCI\_CR1)[WAKE]
  - Control Register 3 (eSCI\_CR3)[M2]
- select *parity type*
  - Control Register 1 (eSCI\_CR1)[PT]

### 42.5.5.2 Transmitter

The transmitter supports the transmission of all frame types defined in [Table 42-19](#), of all break characters defined in [Table 42-21](#), and of all idle characters defined in [Table 42-22](#).

#### 42.5.5.2.1 Transmitter States and Transitions

The transmitter has four basic states which are shown and described in [Table 42-26](#). The state transitions that can be triggered by the application commands are shown in [Table 42-27](#). The state transitions that can be triggered by the module are shown [Table 42-28](#). The state diagram of the transmitter is shown in [Figure 42-26](#).

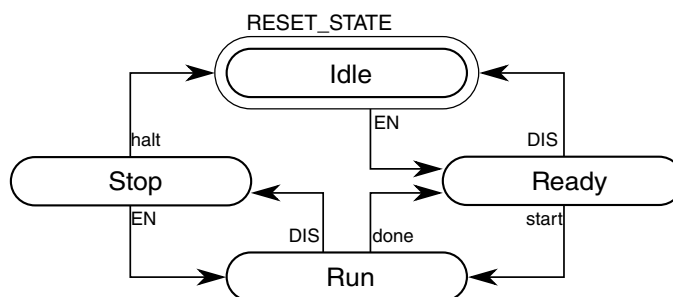


Figure 42-26. Transmitter State Diagram

The current state of the transmitter can be determined by the TE control bit in the Control Register 1 (eSCI\_CR1) and the TACT status bit in Interrupt Flag and Status Register 1 (eSCI\_IFSR1).

Table 42-26. Transmitter States

| State | Indication   |                  | Description                                                          |
|-------|--------------|------------------|----------------------------------------------------------------------|
|       | eSCI_CR1[TE] | eSCI_IFSR1[TACT] |                                                                      |
| Idle  | 0            | 0                | Transmitter is <i>disabled</i> and <i>no</i> transmission is running |
| Ready | 1            | 0                | Transmitter is <i>enabled</i> and <i>no</i> transmission is running  |
| Run   | 1            | 1                | Transmitter is <i>enabled</i> and transmission is running            |
| Stop  | 0            | 1                | Transmitter is <i>disabled</i> and transmission is running           |

The application triggers a transition described in Table 42-27 when it issues a command by writing to the TE bit in the Control Register 1 (eSCI\_CR1). The transition is triggered only if the conditions are fulfilled. As a result of the transition the state of the transmitter is changed as shown in Figure 42-26 and the action given in Table 42-27 is executed.

Table 42-27. Transmitter Application Transitions

| Transition | Command         | Precondition   | Action  | Description                                           |
|------------|-----------------|----------------|---------|-------------------------------------------------------|
| EN         | eSCI_CR1[TE]:=1 | eSCI_CR1[TE]=0 | iPRE:=1 | Transmitter is <i>enabled</i> by application command. |
| DIS        | eSCI_CR1[TE]:=0 | eSCI_CR1[TE]=1 |         | Transmitter is <i>disabled</i> by application command |

The module transition shown in Table 42-28 are triggered when the described condition or event occurs. The send break bit SBK in the Control Register 1 (eSCI\_CR1) is checked for the start condition. The internal commit bit iCMT, the transmitter active bit TACT in the Interrupt Flag and Status Register 1 (eSCI\_IFSR1), the TDRE, and the TC flag in the Interrupt Flag and Status Register 1 (eSCI\_IFSR1) are changed as a action result of the transition.

**Table 42-28. Transmitter Module Transitions**

| Transition | Condition                                              | Action                                          | Description                                                                                                                                                |
|------------|--------------------------------------------------------|-------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| start      | (State=Ready)<br>and<br>(SBK=1 or iPRE=1 or<br>iCMT=1) | TACT:=1                                         | Start of transmission of data frame or special character when data are available or character transmission request is pending.                             |
| done       | State=Run<br>and<br>last stop bit transmitted          | TACT:=0<br>TC:=<br>(SBK=0 & iPRE=0 &<br>iCMT=0) | Finished transmission of data frame or special character and transmitter still enabled.<br><br>Transmission is complete if no transmit request is pending. |
| halt       | State=Stop<br>and<br>last stop bit transmitted         | TACT:=0<br>TC:=1<br>iCMT:=0                     | Finished transmission of data frame or special character and transmitter was disabled.                                                                     |

#### 42.5.5.2.2 Frame and Character Transmission

The transmitter starts the transmission of a data frame or special character when the condition for the *start* transition as described in [Table 42-28](#) is fulfilled. There are three source for data or character transmission. The priority among these source are specified in the following table. All three sources can be available at one point in time.

**Table 42-29. Transmit Source Priority**

| Priority    | Indication      | Transmission Source                 |
|-------------|-----------------|-------------------------------------|
| (highest) 0 | iPRE=1          | Preamble.                           |
| 1           | eSCI_CR1[SBK]=1 | Break character.                    |
| (lowest) 2  | iCMT=1          | SCI Data Register (eSCI_SDR) frame. |

#### 42.5.5.2.3 CPU Controlled SCI Data Frame Transmission

The transmission of a data frame is started when the transmitter is in its Ready state and only the commit bit iCMT is set.

As the first step, the content of the SCI Data Register (eSCI\_SDR) is transferred into the internal transmitter shift register. When this transfer is finished, the internal commit bit iCMT is cleared and the transmit data register empty flag TDRE in the Interrupt Flag and Status Register 1 (eSCI\_IFSR1) is set. If the transmit interrupt enable bit TIE in the Control Register 1 (eSCI\_CR1) is also set, the TDRE flag generates a transmitter interrupt request.

The transmitter shift register then shifts a frame out through the TXD output signal, which is prefaced with a start bit and appended with the parity bit, if configured, and the configured number of stop bits.

When the last stop bit has been transmitted and the application has not disabled the transmitter, the transmitter returns to the Ready state via the *done* transition. If no frame or character transmit request is pending, the transfer complete flag TC in the Interrupt Flag and Status Register 1 (eSCI\_IFSR1) is set.

If the application has disabled the transmitter while the frame is transmitted and stop bit has been transmitted, the transmitter goes into the Idle state via the *halt* transition. The transfer complete flag TC in the Interrupt Flag and Status Register 1 (eSCI\_IFSR1) is set and the internal commit bit iCMT is cleared.

#### **42.5.5.2.4 DMA Controlled SCI Data Frame Transmission**

In this mode, the eSCI module handles the generation of Data Frames internally.

When new data is required for transmission, the module generates the transmit DMA request and the DMA controller delivers the required data via write accesses to SCI Data Register (eSCI\_SDR). The write access to the low byte of SCI Data Register (eSCI\_SDR) triggers the transmission of the data. The write access to the high byte of SCI Data Register (eSCI\_SDR) triggers no internal operation.

The application requests the eSCI module to enter this mode by setting the TXDMA bit in the Control Register 2 (eSCI\_CR2). From this point in time, the module starts the generation of DMA requests and frame transmission. Before entering this mode, the application should perform the following actions:

1. Configure the module for SCI mode.
2. Enable the transmitter by setting TE in Control Register 1 (eSCI\_CR1) to 1.
3. Setup the DMA controller channel and provide frame data in system memory

A block diagram which presents an overview of the DMA Controlled Date Frame Transmission is shown in the following figure.



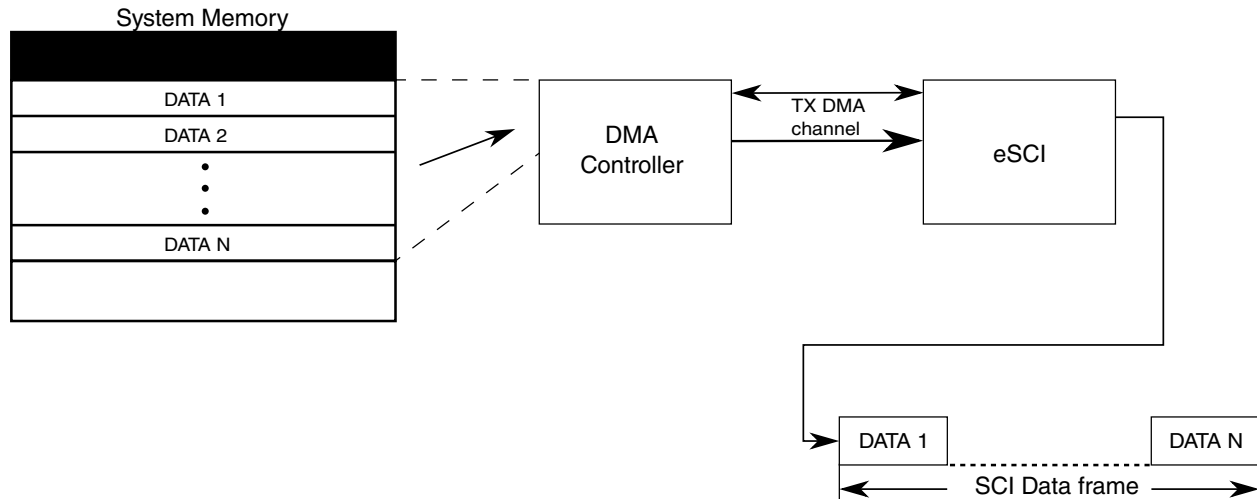


**Figure 42-27. DMA Controlled SCI Data Frame generation**

#### 42.5.5.2.5 Parity Generation

The eSCI module generates the parity bit in transmitted data frame when the parity enable bit PE in the Control Register 1 (eSCI\_CR1) is set. The parity type bit PT in the Control Register 1 (eSCI\_CR1) defines whether the odd or even parity is generated.

#### 42.5.5.2.6 Preamble Transmission

The transmission of a preamble is started when the transmitter is in Ready state, the internal iPRE bit, which is not visible to the application, is set, and the SBK in the Control Register 1 (eSCI\_CR1) is clear.

After the transmission of the stop bit and if the application has not disabled the transmitter, the transmitter returns to the Ready state via the *done* transition. If no frame or character transmit request is pending, the transfer complete flag TC in the Interrupt Flag and Status Register 1 (eSCI\_IFSR1) is set.

If the application has disabled the transmitter while the preamble is transmitted and if the stop bit has been transmitted, the transmitter goes into the Idle state via the *halt* transition. The transfer complete flag TC in the Interrupt Flag and Status Register 1 (eSCI\_IFSR1) is set and the internal commit bit iCMT is cleared.

#### 42.5.5.2.7 Break Character Transmission

The transmission of a break character is started when the transmitter is in Ready state and the send break character bit SBK in the Control Register 1 (eSCI\_CR1) is set. After the transmission of the break character and if the application has not disabled the transmitter,

the transmitter returns to the Ready state via the *done* transition and restarts the transmission. As long as SBK bit remains set, the transmitter continues to send break characters.

When the application has cleared the SBK bit or has disabled the transmitter, the transmitter continues to transmit the current break character and after it has finished the transmission of this break character it transmits a stop bit. The stop bit at the end of a break character sequence guarantees the recognition of the start bit of the next data frame.

After the transmission of the stop bit and if the application has not disabled the transmitter, the transmitter returns to the Ready state via the *done* transition. If no frame or character transmit request is pending, the transfer complete flag TC in the Interrupt Flag and Status Register 1 (eSCI\_IFSR1) is set.

If the application has disabled the transmitter while the break character is transmitted and if the stop bit has been transmitted, the transmitter goes into the Idle state via the *halt* transition. The transfer complete flag TC in the Interrupt Flag and Status Register 1 (eSCI\_IFSR1) is set and the internal commit bit iCMT is cleared.

### 42.5.5.3 Receiver

The receiver supports the reception of all data frame types defined in [Table 42-19](#) and [Table 42-20](#), of all break character defined in [Table 42-21](#), and of all idle characters defined in [Table 42-22](#).

#### 42.5.5.3.1 Receiver States and Transitions

The receiver has four basic states which are shown and described in [Table 42-30](#). The state transitions that can triggered by the application commands are shown in [Table 42-31](#). The state transitions that can triggered by the module are shown in [Table 42-32](#). The state diagram of the transmitter is shown in the following figure.

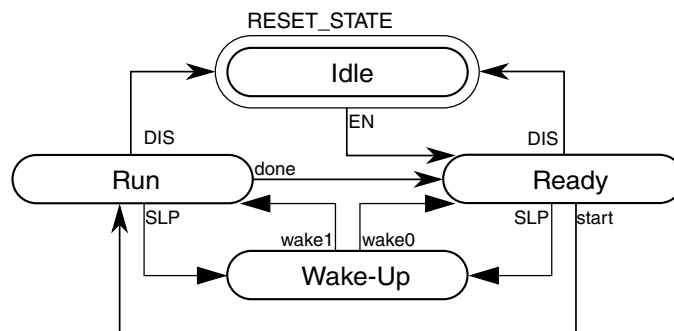


Figure 42-28. Receiver State Diagram

The current state of the receiver can be determined by the RE and RWU bit in the Control Register 1 (eSCI\_CR1) and the RACT status bit in Interrupt Flag and Status Register 1 (eSCI\_IFSR1).

**Table 42-30. Receiver States**

| State   | Indication |      |     | Description                                                    |
|---------|------------|------|-----|----------------------------------------------------------------|
|         | RE         | RACT | RWU |                                                                |
| Idle    | 0          | 0    | 0   | Receiver is <i>disabled</i> and <i>no</i> reception is running |
| Ready   | 1          | 0    | 0   | Receiver is <i>enabled</i> and <i>no</i> reception is running  |
| Run     | 1          | 1    | 0   | Receiver is <i>enabled</i> and reception is running            |
| Wake-up | 1          | -    | 1   | Receiver is in wake-up mode                                    |

The application triggers a transition described in [Table 42-31](#) when it issues a command by writing to the RE bit in the Control Register 1 (eSCI\_CR1). The transition is triggered only if the conditions are fulfilled. As a result of the transition, the state of the receiver is changed as shown in the preceeding figure and the action given in [Table 42-31](#) is executed.

**Table 42-31. Receiver Application Transition**

| Transition | Command | Condition | Action | Description                                        |
|------------|---------|-----------|--------|----------------------------------------------------|
| EN         | RE:=1   | RE=0      |        | Receiver is <i>enabled</i> by application command. |
| DIS        | RE:=0   | RE=1      |        | Receiver is <i>disabled</i> by application command |
| SLP        | RWU:=1  | RE=1      |        | Receiver is set into wake-up mode                  |

The module transitions shown in [Table 42-32](#) are triggered when the described event occurs.

**Table 42-32. Receiver Module Transition**

| Transition | Condition                                                                 | Action  | Description                                          |
|------------|---------------------------------------------------------------------------|---------|------------------------------------------------------|
| start      | (State=Ready,Run)<br>and<br>(start bit qualified)                         | RACT:=1 | Start of reception of data frame or break character. |
| done       | (State=Run)<br>and<br>(start bit not verified or idle character received) | RACT:=0 | Start Bit not Verified or Idle Character received.   |
| wake0      | (State=Wake-up)<br>and<br>(idle character received)                       | RWU:=0  | Wake-up Idle Character received.                     |
| wake1      | (State=Wake-up)                                                           | RWU:=0  | Wake-up address frame received.                      |

Table 42-32. Receiver Module Transition

| Transition | Condition                       | Action | Description |
|------------|---------------------------------|--------|-------------|
|            | and<br>(address frame received) |        |             |

42.5.5.3.2 Receiver Input Mode Selection

This section describes the three receiver input modes supported by the eSCI module. The modes are selected by the LOOPS and RSRC control bits in the Control Register 1 (eSCI\_CR1).

42.5.5.3.3 Dual Wire Mode

In Dual Wire Mode, the eSCI uses the TXD pin for transmitting and the RXD pin for data receiving.

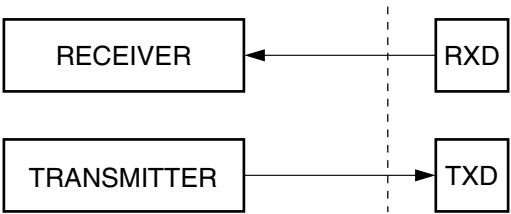


Figure 42-29. Dual Wire Mode

42.5.5.3.4 Single Wire Mode

In Single Wire Mode, the RXD pin is disconnected from the eSCI module and the TXD pin is used for both receiving and transmitting.

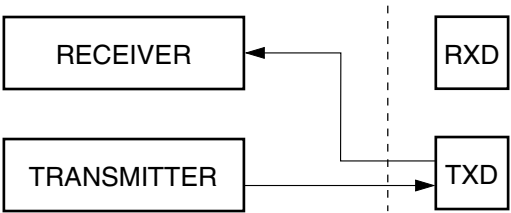


Figure 42-30. Single Wire Mode

In this device, the TXDIR bit (eSCI\_CR2[TXDIR]) does not determine whether the TXD pin is going to be used as an input or an output. The relevant pad control register in the System Integration Unit must be programmed for input or output.

### 42.5.5.3.5 Loop Mode

In Loop Mode, the input of the receiver is driven by the output of the transmitter. The RXD pin is disconnected from the eSCI module.

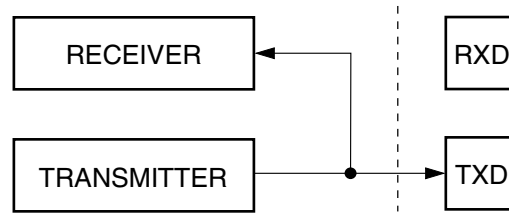


Figure 42-31. Loop Mode

### 42.5.5.3.6 Frame and Character Reception

The receiver is started when it is in Ready or Wake-up state and on the selected receiver input (see [Receiver Input Mode Selection](#)) an active signal is sampled. The receiver enters the Run or Wake-up state. The received bits are recovered by the bit sampling described in [Bit Sampling](#). During the reception, the received bits are shifted into the internal shift register.

### 42.5.5.3.7 Break Character Detection

The receiver does not provide any means to detect the reception of a break character. Instead, break characters are processed as data frames. Due to the received 0 at the stop bit location, the reception of a break character causes at least a framing error. The error reporting is performed as described in [Reception Error Reporting](#).

### 42.5.5.3.8 Idle Character Detection

The Idle character detection starts after the reception of the last stop bit.

### 42.5.5.3.9 CPU Controlled SCI Data Frame Reception

This section describes the reception process when the receiver is in the Run state.

When the required number of frame bits have been received, the payload bits of the received frame are transferred into SCI Data Register (eSCI\_SDR) if the RDRF flag is 0. The receive data register full flag RDRF in Interrupt Flag and Status Register 1 (eSCI\_IFSR1) is set. If the receive interrupt enable bit RIE in the Control Register 1 (eSCI\_CR1) is set, the RDRF interrupt request is generated.

If an idle character has been detected, the IDLE flag in the Interrupt Flag and Status Register 1 (eSCI\_IFSR1) is set. If the idle line interrupt enable bit ILIE in the Control Register 1 (eSCI\_CR1) is set, the IDLE interrupt request is generated.

If any of the receiver errors described in [Reception Error Reporting](#) have occurred, that corresponding flags will be set.

If the application is disabled the receiver, by clearing the receiver enable bit RE in the Interrupt Flag and Status Register 1 (eSCI\_IFSR1), the current frame is discarded and no flags will be updated.

#### **42.5.5.3.10 DMA Controlled SCI Data Frames Reception**

In this mode, the eSCI module controls the reception of SCI Data frames automatically and utilizes the connected DMA channels. A block diagram which presents an overview of the DMA Controlled SCI Data Frame reception is shown in the following figure. The RX DMA channel is used to transfer the received frame data into the memory.

When new data is received, the module generates the receive DMA request and the DMA controller retrieves the provided data from the SCI Data Register (eSCI\_SDR). The read access from the low byte of the SCI Data Register (eSCI\_SDR) signals the end of the DMA cycle for the current data and triggers the reception of new data. The read access from the SCI Data Register (eSCI\_SDR) triggers no internal action.

The application requests the eSCI module to enter this mode by setting the RXDMA bit in the Control Register 2 (eSCI\_CR2). From this point in time, the module start the generation of DMA requests and frame transmission and reception. Before entering this mode, the application should perform the following actions:

1. Configure the module for SCI mode.
2. Enable the receiver by setting RE in Control Register 1 (eSCI\_CR1) to 1.
3. Setup the DMA controller channel.

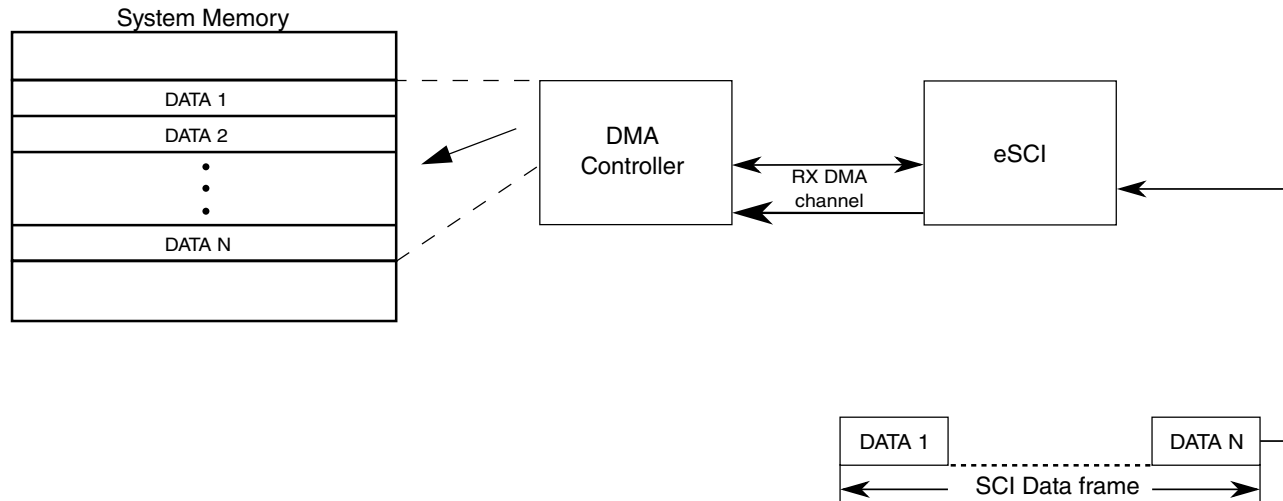


**Figure 42-32. DMA Controlled SCI Data Frame Reception**

#### 42.5.5.3.11 Receiver Overrun

When the eSCI module has received a frame and attempts to transfer the payload data of the received frame into the SCI Data Register (eSCI\_SDR) but neither the application nor the DMA controller has SCI Data Register (eSCI\_SDR) since its last update, the overrun flag OR in the Interrupt Flag and Status Register 1 (eSCI\_IFSR1) is set. The data contained in SCI Data Register (eSCI\_SDR) are not changed and the received data are lost.

#### 42.5.5.3.12 Wake-up Frame Reception

This section describes the reception process when the receiver is in the Wake-up state.

When the required number of frame bits have been received, the payload bits of the received frame are transferred into SCI Data Register (eSCI\_SDR) if the RDRF flag is 0.

If the *address-mark* wake-up mode is selected and the received frame has the address bit set, the receive data register full flag RDRF in Interrupt Flag and Status Register 1 (eSCI\_IFSR1) is set. If the receive interrupt enable bit RIE in the Control Register 1 (eSCI\_CR1) is set, the RDRF interrupt request is generated. The RWU bit is cleared, and the receiver enters the Run state via the wake1 transition.

If the *idle line* wake-up mode is selected and the receiver has detected an idle character, The RWU bit is cleared, and the receiver enters the Ready state via the wake0 transition.

If any of the receiver errors described in [Reception Error Reporting](#) have occurred, the corresponding flags will be set.

### 42.5.5.3.13 Bit Sampling

The receiver samples the selected receiver input (see [Receiver Input Mode Selection](#)) with the receiver clock RCLK. The bit sampling for start bit detection is shown in [Figure 42-33](#), the bit sampling for data and stop bit reception is shown in [Figure 42-34](#). The samples indicated by dashed arrows are not used by the receiver. The received data bits are transferred into the internal shift register after the data strobing. If noise or framing errors are detected, this is flagged as described in [Reception Error Reporting](#).

### 42.5.5.3.14 Bit Synchronization

To adjust for baud rate mismatch, a synchronization of the cyclic receive sample counter RSC is performed during start bit reception as described in [Start Bit Sampling](#).

### 42.5.5.3.15 Start Bit Sampling

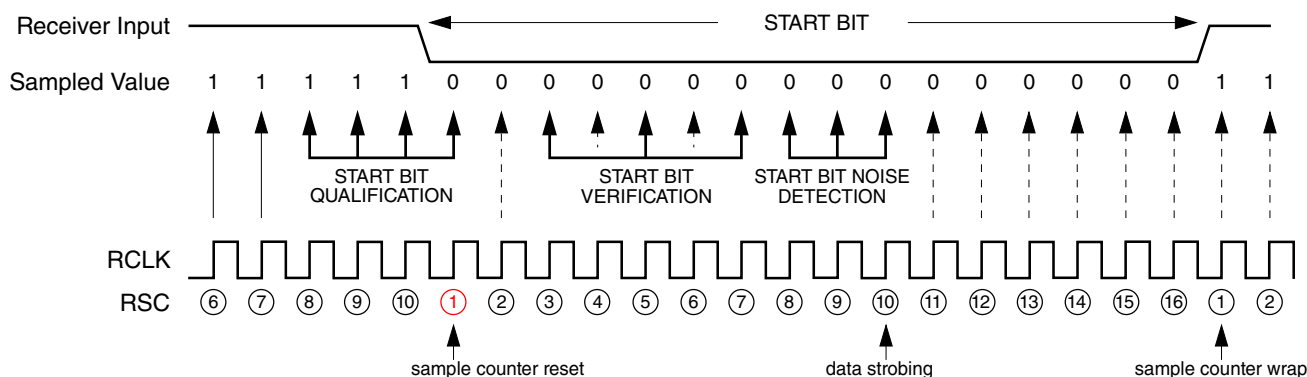


**Figure 42-33. Start Bit Sampling and Strobing**

The sampling of the start bit consists of three phases, the start bit qualification, the start bit verification, and the start bit noise detection.

#### 42.5.5.3.15.1 Start Bit Qualification

To adjust for baud rate mismatch, the cyclic receive sample counter RSC is re-synchronized after a successful start bit qualification.

A start bit is successfully qualified if

- the start qualification is active, and
- a low sample is read, and
- the low sample was preceded by three consecutive high samples.

The start bit qualification becomes active



- after module reset, or
- after receiver disable and subsequent enable, or
- after the 7-th sample if the start bit verification failed, or
- after the 10-th sample of last stop bit of the preceding frame (example shown in [Figure 42-33](#)).

The start bit qualification becomes inactive

- after successful start bit qualification.

#### 42.5.5.3.15.2 Start Bit Verification

After the successful start bit qualification the receiver starts to verify the start bit by a two out of three samples majority voting.

A start bit is verified if at least two out of the three sample RSC3, RSC5, and RSC7 are sampled low. Noise is detected when exactly one out of the three samples is high. In this case, the noise flag eSCI\_IFSR1[NF] is set. The result of the start bit verification is summarized in [Table 42-33](#).

**Table 42-33. Start Bit Verification Result**

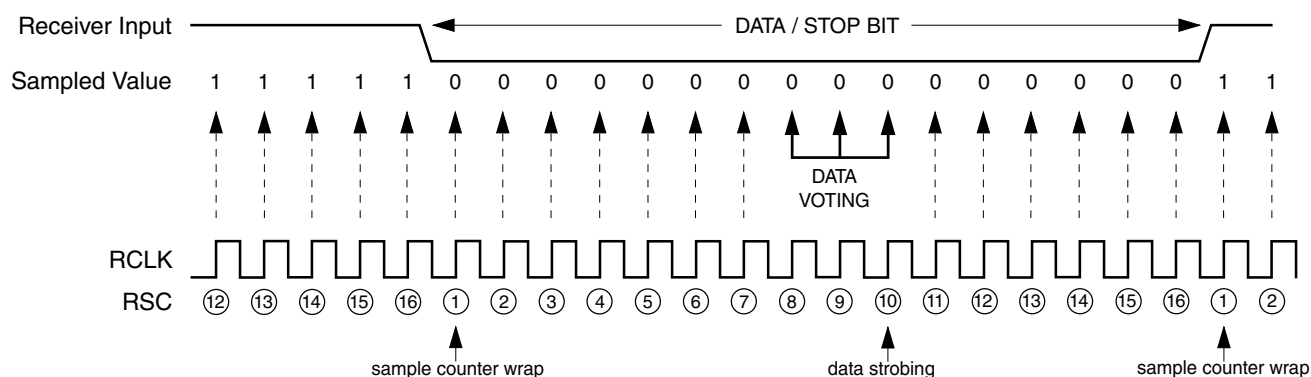
| [RSC3, RSC5, RSC7] | Start Bit Verified | Verification Noise Detected |
|--------------------|--------------------|-----------------------------|
| 000                | Yes                | No                          |
| 001                | Yes                | Yes                         |
| 010                | Yes                | Yes                         |
| 100                | Yes                | Yes                         |
| 011                | No                 | No                          |
| 101                | No                 | No                          |
| 110                | No                 | No                          |
| 111                | No                 | No                          |

If the start bit verification was *not successful*, the receiver activates the start bit qualification. If the start bit verification was *successful*, the receiver continues sampling to perform data noise detection on the samples at RSC8, RSC9, and RSC10. The result of the start bit data noise detection is summarized in [Table 42-34](#). If noise is detected, the noise flag eSCI\_IFSR1[NF] is set.

**Table 42-34. Start Bit Noise Detection**

| [RSC8, RSC9, RSC10] | Noise Detected |
|---------------------|----------------|
| 000                 | No             |
| 001                 | Yes            |
| 010                 | Yes            |
| 100                 | Yes            |
| 011                 | Yes            |
| 101                 | Yes            |
| 110                 | Yes            |
| 111                 | Yes            |

#### 42.5.5.3.16 Data Bit Sampling

**Figure 42-34. Data and Stop Bit Sampling and Strobming**

To determine the value of a data bit and to detect noise, a two out of three majority voting is performed on the samples RSC8, RSC9, and RSC10. The following table summarizes the results of the data bit sample. The receiver detects the number of data bit according to the selected frame format. If noise is detected, the noise flag eSCI\_IFSR1[NF] is set.

**Table 42-35. Data Bit Sampling**

| [RSC8, RSC9, RSC10] | Data Bit Value | Noise Detected |
|---------------------|----------------|----------------|
| 000                 | 0              | No             |
| 001                 | 0              | Yes            |
| 010                 | 0              | Yes            |
| 100                 | 0              | Yes            |
| 011                 | 1              | Yes            |
| 101                 | 1              | Yes            |
| 110                 | 1              | Yes            |
| 111                 | 1              | No             |

### 42.5.5.3.17 Stop Bit Verification

The reception of a valid stop bit is verified if at least two out of the sample RSC8, RSC9, and RSC10 are sampled high. If this is not the case, a framing error is detected. Noise is detected if not all of the samples are of the same value. In this case, the noise flag eSCI\_IFSR1[NF] is set. The result of the stop bit verification is summarized in the following table.

**Table 42-36. Stop Bit Verification**

| [RSC8, RSC9, RSC10] | Stop Bit Verified | Framing Error Detected | Noise Detected |
|---------------------|-------------------|------------------------|----------------|
| 000                 | No                | Yes                    | No             |
| 001                 | No                | Yes                    | Yes            |
| 010                 | No                | Yes                    | Yes            |
| 100                 | No                | Yes                    | Yes            |
| 011                 | Yes               | No                     | Yes            |
| 101                 | Yes               | No                     | Yes            |
| 110                 | Yes               | No                     | Yes            |
| 111                 | Yes               | No                     | No             |

### 42.5.5.3.18 Parity Checking

The eSCI module calculates the parity of a received character and checks it versus the received parity bit in the received data frame when the parity enable bit PE in the Control Register 1 (eSCI\_CR1) is set. The parity type bit PT in the Control Register 1 (eSCI\_CR1) defines whether to check for odd or even parity generation. If a parity error is detected, it is reported as described in [Reception Error Reporting](#).

## 42.5.5.4 Reception Error Reporting

The receiver can detect four error types: parity errors, framing errors, noise errors, and the overrun error.

The receiver reports the errors detected during frame reception at the end of the reception of the last stop bit of a frame. For error reporting the receiver utilizes the OR, NF, FE, and PF flags in the Interrupt Flag and Status Register 1 (eSCI\_IFSR1).

If the receiver has detected an overrun as described in [Receiver Overrun](#), only the OR flag is set. All other error flags are not updated.

If the receiver has detected noise as described in [Bit Sampling](#), the NF flag is set.

If the receiver has *not* detected an overrun and has detected a framing error as described in [Bit Sampling](#), the FE flag is set.

If the receiver has *not* detected an overrun and has detected a parity error as described in [Parity Checking](#), the PF flag is set.

### 42.5.5.5 Multiprocessor Communication

The multiprocessor communication allows one processor to send blocks of frames to other processors on the same serial link. To avoid the received data interrupt for frames not intended for the processor, the eSCI receiver can be put into the Wake-up state. If the receiver is in the Wake-up state, the eSCI will still load the received data into the SCI Data Register (eSCI\_SDR), but will not set the RDRF flag and consequently not request the RDRF interrupt.

The receiver leaves the Wake-up state and clears the RWU bit in the Control Register 1 (eSCI\_CR1) when the wake-up pattern configured by WAKE bit in Control Register 1 (eSCI\_CR1) is received. The eSCI module supports two types of wake-up patterns, the idle-line wakeup pattern and the address-mark wake-up pattern.

#### 42.5.5.5.1 Idle-Line Wake Up

The idle-line wake-up mode is selected when the WAKE bit in Control Register 1 (eSCI\_CR1) is 0. In this mode, the receiver leaves the wake-up state when an idle character is detected as described in [Idle Character Detection](#). The next received frame is the address frame that contains address information which can be evaluated by the application. If the application decides not to receive the frame block, it can set the RWU bit in the Control Register 1 (eSCI\_CR1) and return the receiver to the wake-up state.

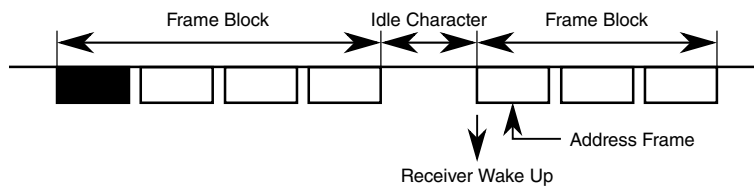


Figure 42-35. Idle-Line Wake Up

#### 42.5.5.5.2 Address-Mark Wake Up

The address-mark wake-up mode is selected when the WAKE bit in Control Register 1 (eSCI\_CR1) is 1. If the WAKE bit is set, the address bit is added to the frame format. In this mode, the receiver leaves the wake-up state when a data frame with the address bit value of 1 was received. This frame is the address frame and contains address

information which can be evaluated by the application. If the application decides not to receive the frame block, it can set the RWU bit in the Control Register 1 (eSCI\_CR1) and return the receiver to the wake-up state. All data frames that belong to the frame block must have the address bit cleared.

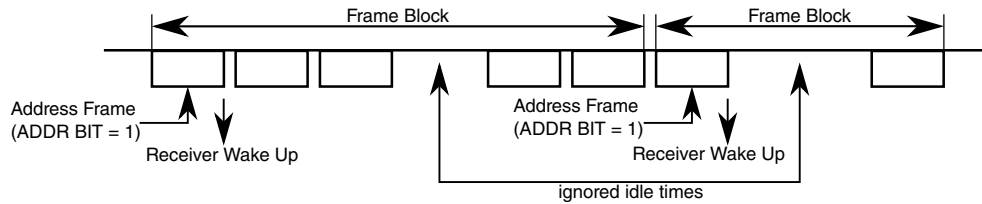


**Figure 42-36. Address-Mark Wake Up**

## 42.5.6 LIN Mode

The eSCI provides support for the LIN protocol. It can be used to automate most tasks of a LIN master. In conjunction with the DMA interface it is possible to transmit entire LIN frames and sequences of LIN frames as well as to receive data from LIN slaves without application intervention. There is no special support for LIN slave mode.

### 42.5.6.1 LIN Mode Configuration

The application must configure the following bits and fields in order to achieve correct LIN operation. The configuration of bits and fields not mentioned in this section depend on the connected LIN slaves and the current application.

- enable *LIN* Mode
  - LIN Control Register 1 (eSCI\_LCR1)[LIN]:= 1
- select *RXD* pin as receiver input
  - Control Register 1 (eSCI\_CR1)[LOOPS]:= 0
  - Control Register 1 (eSCI\_CR1)[RSRC]:= 0
- select *LIN byte fields* as used frame format
  - Control Register 1 (eSCI\_CR1)[M]:= 0
  - Control Register 1 (eSCI\_CR1)[PE]:= 0
  - Control Register 1 (eSCI\_CR1)[WAKE]:= 0
  - Control Register 3 (eSCI\_CR3)[M2]:= 0

- select *break character length* of 13 bit as required by LIN 2.0
  - Control Register 2 (eSCI\_CR2)[BRCL]:= 1
- select transmission stop on *bit error* detection
  - Control Register 2 (eSCI\_CR2)[BESTP]:= 1
- select transmission DMA stop on *bit error* detection
  - Control Register 2 (eSCI\_CR2)[BSTP]:= 1
- enable both *transmitter* and *receiver*
  - Control Register 1 (eSCI\_CR1)[TE]:= 1
  - Control Register 1 (eSCI\_CR1)[RE]:= 1

## 42.5.6.2 LIN frame formats

The term LIN frame refers to a sequence of LIN byte fields preceded by a break character, both are described in [Frame Formats](#). The eSCI module allows to generate LIN frames for LIN slaves of LIN standards 1.3 and 2.0.

### 42.5.6.2.1 LIN byte field reception

The reception of a LIN byte field starts with the successful start bit qualification and is finished with the reception of the 16-th sample of the stop bit when no start bit qualification pattern has been detected. If a start bit qualification pattern has been detected at or after the 10-th sample of the stop bit, the reception ends at this sample. An ongoing reception is indicated by the RACT status bit in Interrupt Flag and Status Register 1 (eSCI\_IFSR1).

The RACT flag is set if all of the following conditions are fulfilled:

1. the receiver is enabled (eSCI\_CR1[RE] = 1), and
2. the LIN task is not in reset (eSCI\_LCR1[LRES] = 0), and
3. the start bit qualification pattern has been received (see [Start Bit Sampling](#)).

The RACT flag is cleared if at least one of the following conditions is fulfilled,

1. the receiver is disabled (eSCI\_CR1[RE] = 0), or
2. the LIN task is in reset (eSCI\_LCR1[LRES] = 1), or

3. the start bit verification fails at sample 7 according to [Table 42-33](#), or
4. the 16-th sample of the stop bit has been received and no start bit qualification pattern has been detected at or after the 10-th sample.

#### 42.5.6.2.2 Standard LIN frames

A standard LIN frame, shown in the following figure consists of a break character, a sync field, an ID field, zero or more data fields, and a checksum field. The data fields and the checksum field are generated by the LIN master for TX LIN frames and generated by the LIN slave for RX LIN frames. The header fields will always be generated by the LIN master.



**Figure 42-37. Standard LIN frame format**

#### 42.5.6.2.3 CRC Enhanced LIN frames

The CRC Enhanced LIN frames shown in the following figure contain two additional CRC byte fields. These fields are located between the last data field and the Checksum field. The value of the CRC is calculated on the same byte fields as the Checksum is calculated on. The polynomial used for the CRC calculation is defined by LIN CRC Polynomial Register (eSCI\_LPR). The eSCI module generates the CRC fields for TX frames and checks the CRC fields for RX frames if the CRC bit in the LIN Transmit Register (eSCI\_LTR) was written with a value of 1.



**Figure 42-38. CRC Enhanced LIN frame format**

The CRC Enhanced LIN frames are not part of the LIN standard.

#### 42.5.6.3 LIN TX Frame generation

The eSCI module supports two modes of LIN TX Frame generation, the CPU controlled mode and the DMA controlled mode. In the CPU controlled mode, the application provides the required frame configuration and frame data by subsequent CPU write accesses to the LIN Transmit Register (eSCI\_LTR). In the DMA controlled mode, the DMA controller provides the required frame configuration and frame data in response to DMA requests generated by the eSCI module.

### 42.5.6.3.1 CPU Controlled LIN TX Frame generation

In this mode, the application initiates the generation of an LIN TX Frame and provides the data to be transmitted by a sequence of subsequent CPU write accesses to the LIN Transmit Register (eSCI\_LTR). When the eSCI module has processed the data written into LIN Transmit Register (eSCI\_LTR), the TXRDY interrupt flag in the Interrupt Flag and Status Register 2 (eSCI\_IFSR2) will be set.

The application should clear the TXRDY interrupt flag *before* writing data into the LIN Transmit Register (eSCI\_LTR) because the eSCI module will set the TXRDY one clock cycle after the write access.

The first data written to the LIN Transmit Register (eSCI\_LTR) provides the Identifier and Identifier Parity fields. The second data written defines the number of data bytes to be transmitted. The third data written defines the CRC and checksum generation. The TD bit has to set to 1 in order to invoke the LIN TX frame generation. The value of the TO field is ignored by the eSCI module for LIN TX frames.

After the third data was written the generation of a LIN TX frame is started. Firstly, a break field is transmitted, then the synch field and the protected identifier field.

All subsequent write accesses to the LIN Transmit Register (eSCI\_LTR) provide data bytes to be transmitted via the LIN bus. A data byte field will be transmitted as soon as data are available. After the last data byte, defined by the value written to the LEN field, was send out, the configured CRC and checksum fields will be send out.

After the transmission of the checksum field of the LIN TX frame, the write access counter for the LIN Transmit Register (eSCI\_LTR) is reset and the FRC interrupt flag in the Interrupt Flag and Status Register 2 (eSCI\_IFSR2) is set.

### 42.5.6.3.2 DMA Controlled LIN TX Frame generation

In this mode, the eSCI module controls the generation of an LIN TX Frame. When new data is required for transmission, the eSCI module generates the transmit DMA request and the DMA controller delivers the required data. The application requests the eSCI module to enter this mode by setting the TXDMA bit in the Control Register 2 (eSCI\_CR2). From this point in time, the module starts the generation of DMA requests and initiates the frame transmission. Before entering this mode, the application should perform the following actions:

1. Configure the module for LIN mode.
2. Enable the transmitter by setting TE in Control Register 1 (eSCI\_CR1) to 1.
3. Setup the DMA controller channel and provide frame data in system memory.



A block diagram which presents an overview of the DMA Controlled LIN TX Frame is shown in the following figure. The content of the fields in the memory is the same as described in LIN Transmit Register (eSCI\_LTR) - LIN TX frame generation.

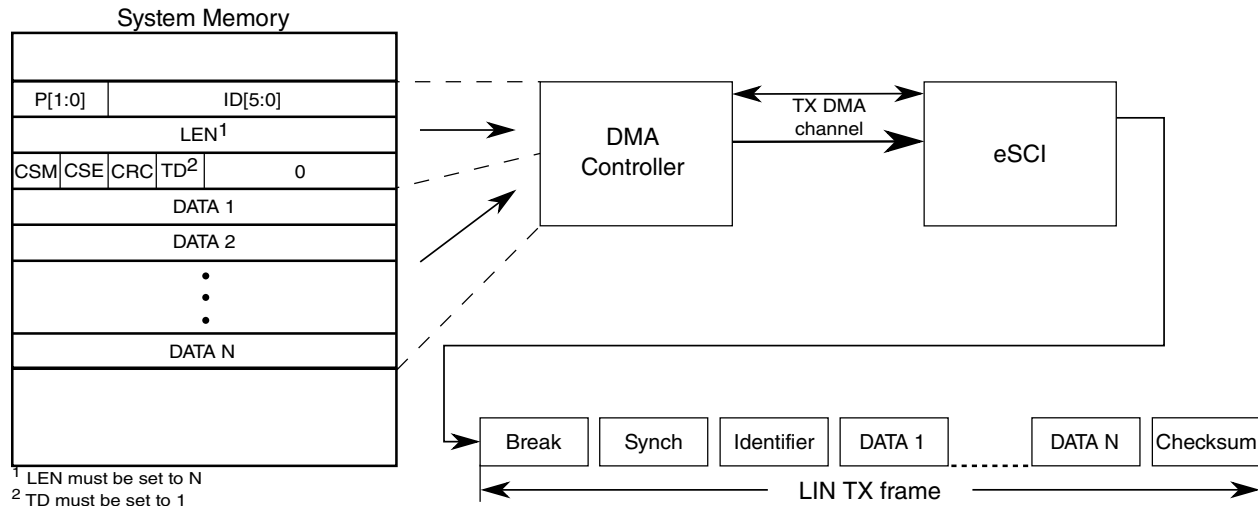


**Figure 42-39. DMA Controlled LIN TX Frame generation**

#### 42.5.6.4 LIN RX frame generation

The eSCI module supports two modes of LIN RX Frame generation and reception, the CPU controlled mode and the DMA controlled mode. In the CPU controlled mode, the application provides the required data by subsequent CPU write accesses to the LIN Transmit Register (eSCI\_LTR) and retrieves the received data by subsequent CPU read accesses to the LIN Receive Register (eSCI\_LRR). In the DMA controlled mode, the DMA controller provides the required frame configuration data in response to DMA requests generated by the eSCI module and transfers the received frame data to the memory in response to DMA requests generated by the eSCI module.

##### 42.5.6.4.1 CPU Controlled LIN RX Frames generation

In this mode, the application initiates the generation of a LIN RX Frame by a sequence of subsequent CPU write accesses to the LIN Transmit Register (eSCI\_LTR). When the eSCI module has processed the data written into LIN Transmit Register (eSCI\_LTR), the TXRDY interrupt flag in the Interrupt Flag and Status Register 2 (eSCI\_IFSR2) will be set.

The application should clear the TXRDY interrupt flag *before* writing data into the LIN Transmit Register (eSCI\_LTR) because the eSCI module will set the TXRDY one clock cycle after the write access.

The first data written to the LIN Transmit Register (eSCI\_LTR) provides the Identifier and Identifier Parity fields. The second data written defines the number of data bytes requested from the LIN slave. The third data written defines the CRC and checksum generation. The TD bit has to set to 0 to invoke the RX frame generation. The TO field defines the upper part of the timeout value. The fourth byte written defines the lower part of the timeout value.

After the fourth byte was written the generation of a LIN RX frame is started. Firstly, a break field is transmitted, then the synch field and the protected identifier field. After the transmission of the protected identifier, the eSCI module starts to receive the frame data transmitted by the LIN slave. When the module has received a complete byte field, the received data are transferred into the LIN Receive Register (eSCI\_LRR) and the receive data ready flag RXRDY in the Interrupt Flag and Status Register 2 (eSCI\_IFSR2) is set.

The application can retrieve the received data by subsequent read access from LIN Receive Register (eSCI\_LRR) after checking the RXRDY flag. The application should clear the RXRDY flag immediately after reading the LIN Receive Register (eSCI\_LRR).

After the reception of the configured number of data from the slave, the module starts the reception of the configured CRC and Checksum byte fields. These data are not transferred into the LIN Receive Register (eSCI\_LRR). The CRC and Checksum checking is performed internally. In case of errors, they will be reported as described in [LIN Error Reporting](#).

After the reception of the checksum field of the LIN RX frame, the FRC interrupt flag in the Interrupt Flag and Status Register 2 (eSCI\_IFSR2) is set.

#### 42.5.6.4.2 DMA Controlled LIN RX Frames generation

In this mode, the eSCI module controls the generation of LIN RX frame header and the reception of the frame data automatically and utilizes the two connected DMA channels. A block diagram which presents an overview of the DMA Controlled LIN RX Frame generation and reception is shown in [Figure 42-40](#). The content of the header fields in the memory is the same as described in LIN Transmit Register (eSCI\_LTR) - LIN RX frame generation. The TX DMA channel is used to fetch the LIN RX frame header and control information. The RX DMA channel is used to transfer the received frame data into the memory.

When new data is required for transmission, the module generates the transmit DMA request and the DMA controller delivers the required data. When new data is received, the module generates the receive DMA request and the DMA controller retrieves the provided data.

The application requests the eSCI module to enter this mode by setting the RXDMA bit in the Control Register 2 (eSCI\_CR2). From this point in time, the module starts the generation of DMA requests and frame transmission and reception. Before entering this mode, the application should perform the following actions:

1. Configure the module for LIN mode.
2. Enable transmitter and receiver by setting TE and RE in Control Register 1 (eSCI\_CR1) to 1.
3. Setup the two DMA controller channels and provide frame header data in system memory.

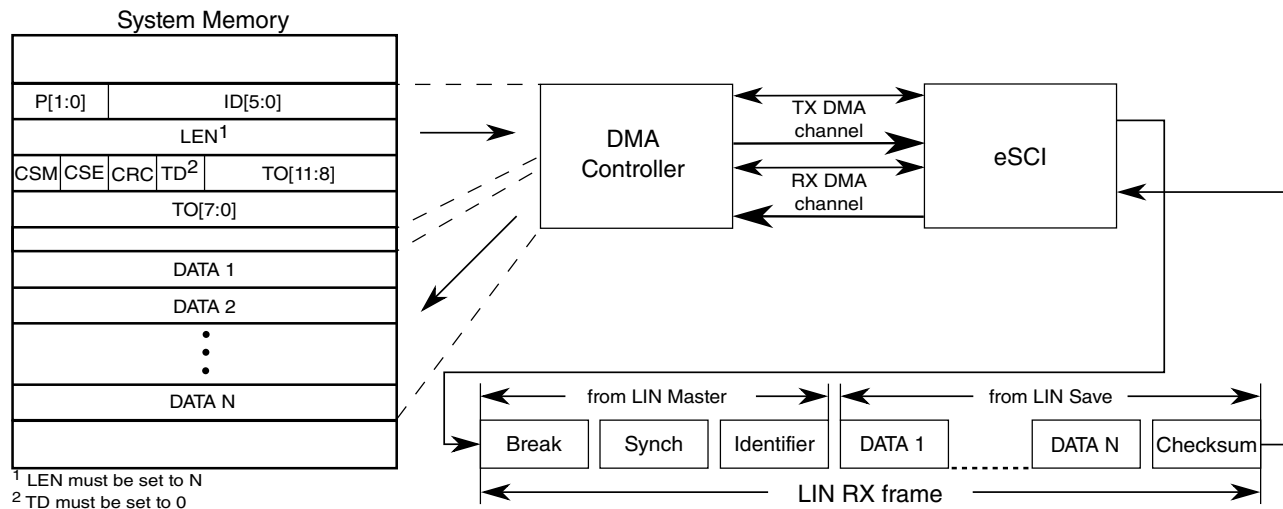


**Figure 42-40. DMA Controlled LIN RX Frame generation and reception**

### 42.5.6.5 LIN Error Reporting

This section describes error checking and the signaling of detected errors in LIN mode.

#### 42.5.6.5.1 Physical Bus Error Detection

If the receiver input is sampled 0 for at least 31 sample clock cycles after the start of the transmission of a LIN frame, the physical bus error flag PBERR in the Interrupt Flag and Status Register 2 (eSCI\_IFSR2) will be set.

### 42.5.6.5.2 Unrequested Activity Detection

If an unrequested byte is received (i.e. a byte which is not part of an RX frame) which is not recognized as a wake-up or break character, the bit error flag BERR in the Interrupt Flag and Status Register 1 (eSCI\_IFSR1) is set. In addition the RXRDY flag will also be set, the LINRX register must be read before normal operations can proceed.

### 42.5.6.5.3 Standard Bit Error Detection

The standard bit error detection is enabled when the fast bit error detection control bit FBR in the Control Register 2 (eSCI\_CR2) is 0. The standard bit error detection is performed after each LIN byte field transmission.

During the transmission of the LIN frame header and LIN frame data, the receiver is running and receives the signal values on the serial bus. After the complete transmission and the related reception of a LIN byte field, the eSCI compares the data that was transmitted and the data that has been received. If they do not match, the bit error interrupt flag BERR in the Interrupt Flag and Status Register 1 (eSCI\_IFSR1) is set.

### 42.5.6.5.4 Fast Bit Error Detection

Fast Bit Error Detection has been designed to allow flagging of LIN bit errors while they occur, rather than flagging them after a byte transmission has completed (see [Figure 42-41](#)).

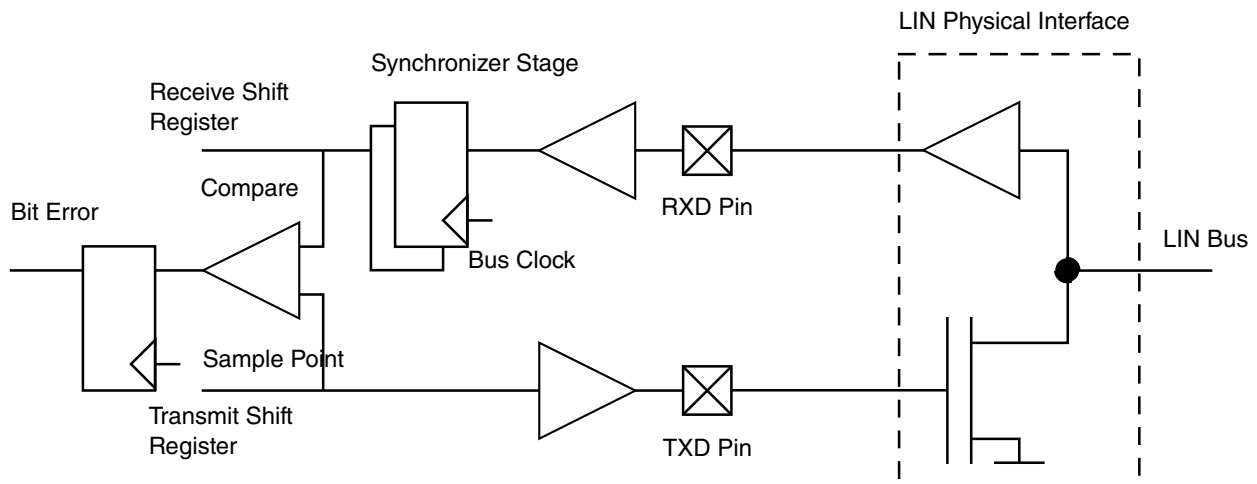


**Figure 42-41. Fast Bit Error Detection on a LIN Bus**

If fast bit error detection bit FBR in the Control Register 2 (eSCI\_CR2) is set, the eSCI will compare the transmitted and the received data stream while the transmitter is active (not idle). Once a mismatch between the transmitted data and the received data is detected, the bit error flag BERR will be set.

To adjust to different bus loads the sample point at which the incoming bit is compared to the one which was transmitted can be selected with the BESM bit in the Control Register 2 (eSCI\_CR2). If eSCI\_CR2[BESM] = 1, the comparison will be performed with sample RS13, otherwise with RS9 (see [Figure 42-42](#)) (also see [Bit Sampling](#)).

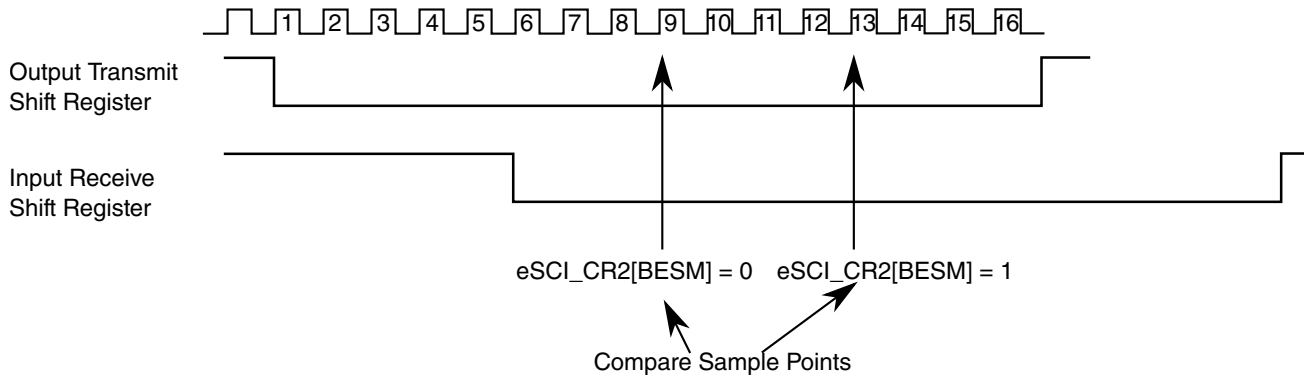


**Figure 42-42. Timing Diagram Fast Bit Error Detection**

### Note

To calculate the exact position of the sample point with regard to the RX pin, the delays through the pads and the two Bus Clock cycle delay through the input synchronizer also needs to be taken into account.

#### 42.5.6.5.5 Slave-Not-Responding-Error Detection

The Slave-Not-Responding-Error is defined in [LIN Specification Package Revision 1.3; December 12, 2002](#); 6 ERROR AND EXCEPTION HANDLING. The LIN specification requires that a NO\_RESPONSE\_ERROR has to be detected if a message frame is not fully completed within the maximum length  $T_{\text{FRAME\_MAX}}$  by any slave task upon transmission of the SYNCH and IDENTIFIER fields. The maximum frame length  $T_{\text{FRAME\_MAX}}$  is defined in [LIN Specification Package Revision 1.3; December 12, 2002](#); 3.3 LENGTH OF MESSAGE FRAME AND BUS SLEEP DETECT, as

$$T_{\text{FRAME\_MAX}} = (10 \cdot N_{\text{DATA}} + 45) \cdot 1.4$$

where  $N_{\text{DATA}}$  is the number of data byte fields of the message frame.

The STO interrupt flag in the Interrupt Flag and Status Register 2 (eSCI\_IFSR2) will be set, if an LIN RX frame was not fully received in the amount of time specified in the timeout value field TO in the LIN Transmit Register (eSCI\_LTR). The time period starts with the falling edge of the transmitted LIN break character and is specified in units of transmit bits.

To achieve LIN compliant Slave-Not-Responding-Error detection, the timeout value TO in the LIN Transmit Register (eSCI\_LTR) field has to be set to  $T_{\text{FRAME\_MAX}}$  when a LIN RX frame is initiated.

#### 42.5.6.5.6 Checksum Error Detection

If the checksum enable bit CSE in the LIN Transmit Register (eSCI\_LTR) was set, the checksum checking is performed based on the received checksum byte. The checksum mode is selected by the CSM bit in the LIN Transmit Register (eSCI\_LTR). If the value received in the checksum bytes did not match the calculated checksum, the checksum error flag CKERR in the Interrupt Flag and Status Register 2 (eSCI\_IFSR2) will be set.

#### 42.5.6.5.7 CRC Error Detection

The CRC checking is performed on the two received CRC bytes CRC1 and CRC2 if the CRC Enhanced LIN frame format was selected by the CRC bit in the LIN Transmit Register (eSCI\_LTR). If the value received in the two CRC bytes did not match the calculated CRC pattern, the CRC error flag CERR in the Interrupt Flag and Status Register 2 (eSCI\_IFSR2) will be set.

#### 42.5.6.5.8 Overflow Detection

When the receiver has received the next byte field, which should be transferred into the LIN Receive Register (eSCI\_LRR), but neither the application nor the RX DMA channel have read data from this register since the last update, the received data overflow flag OVFL in the Interrupt Flag and Status Register 2 (eSCI\_IFSR2) will be set. In this case the content of the LIN Receive Register (eSCI\_LRR) is not changed. The data received most recently are lost.

### 42.5.6.6 LIN Wake Up

The section describes the LIN Wake Up behavior of the eSCI module.

#### 42.5.6.6.1 LIN Wake-Up Request Generation

The eSCI module can cause the LIN bus to exit the sleep mode by sending a wake-up signal frame, which consists of a wake-up signal 0x80 (consisting of 8 dominant bits followed by 1 recessive bit), followed by the wake-up delimiter period as defined by the WUD field in the LIN Control Register 1 (eSCI\_LCR1), followed by Idle time which is  $n$  bits, where  $n \geq 0$ . This Idle time depends on the time until next frame is started by the software.

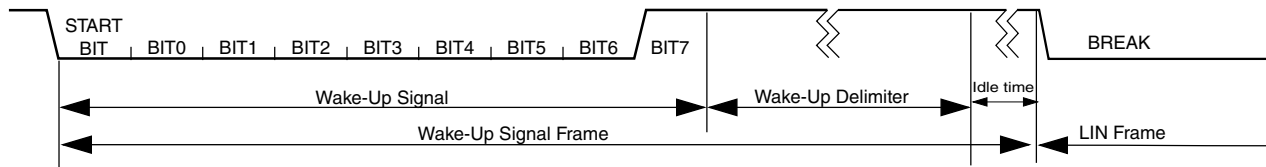


**Figure 42-43. LIN Wake-Up Signal Frame**

The application triggers the transmission of a wake-up signal frame by writing 1 to the LIN bus wake-up trigger WU in the LIN Control Register 1 (eSCI\_LCR1).

The LIN Specification 2.0 requires the generation of LIN wake-up signals as dominant pulses longer than 250  $\mu$ s and shorter than 5 ms. To achieve this, the eSCI module has to be programmed to a baud rate between 32 kBaud and 1.6 kBaud. With each of these baud rate settings, the wake-up signal is transmitted as a dominant pulse longer than 250  $\mu$ s and shorter than 5 ms.

#### 42.5.6.6.2 LIN Wake-Up Request Detection

The eSCI module detects a LIN wake-up requests when

1. one of the characters 0x00, 0x80, or 0xC0 has been received,
2. followed by zero or more low bits,
3. followed by at least one high bit, and
4. no LIN frame transmission or reception is started or running during the reception above

If a LIN wake-up request has been detected, the LIN wake-up flag LWAKE in the Interrupt Flag and Status Register 2 (eSCI\_IFSR2) will be set after the reception of the first high bit.

The LIN Specification 2.0 requires the detection of LIN wake-up requests as dominant pulses longer than 150  $\mu$ s. To achieve this, the eSCI module has to be programmed to the maximum baud rate that is not greater than 43.77 kBaud. With this baud rate setting, any dominant pulse longer than 150  $\mu$ s is decoded as at least 7 dominant bits (one start and 6 data bits) and consequently as one of the characters 0xC0, 0x80, or 0x00.

#### 42.5.6.7 LIN Protocol Engine Stop and Reset

The LIN protocol engine is stopped and reset when the application sets the LRES control bit in the LIN Control Register 1 (eSCI\_LCR1) to 1. In this case, the LIN protocol engine will stop immediately. No new transmissions or receptions are initiated, the LIN serial

bus is driven with the recessive value 1. Additionally to the stop and reset of the LIN protocol engine the receiver and transmitter modules are stopped and reset as well, and the receive and transmit DMA requests are deasserted.

In order to start the LIN Protocol Engine with idle transmitter and receiver processes, the LRES bit should be asserted until all of the status bits DACT, WACT, LACT, TACT, and RACT in the Interrupt Flag and Status Register 1 (eSCI\_IFSR1) are cleared. These status bits are cleared within one bit time after assertion of the LRES bit.

## 42.5.7 Interrupts

This section describes the interrupt sources and interrupt request generation.

### 42.5.7.1 Interrupt Flags and Enables

All interrupt sources, interrupt flags, and interrupt enable bits are listed in the following table. This table indicates the operational modes, where the interrupt flags can be set by the eSCI module.

**Table 42-37. eSCI Interrupt Flags and Interrupt Enable Bits**

| Interrupt Source      | Operational Mode | Interrupt Flag    | Interrupt Enable Bit |
|-----------------------|------------------|-------------------|----------------------|
| Transmitter           | SCI              | eSCI_IFSR1[TDRE]  | eSCI_CR1[TIE]        |
| Transmitter           | SCI, LIN         | eSCI_IFSR1[TC]    | eSCI_CR1[TCIE]       |
| Receiver              | SCI              | eSCI_IFSR1[RDRF]  | eSCI_CR1[RIE]        |
| Receiver              | SCI              | eSCI_IFSR1[IDLE]  | eSCI_CR1[ILIE]       |
| Receiver              | SCI              | eSCI_IFSR1[OR]    | eSCI_CR2[ORIE]       |
| Receiver              | SCI, LIN         | eSCI_IFSR1[NF]    | eSCI_CR2[NFIE]       |
| Receiver              | SCI, LIN         | eSCI_IFSR1[FE]    | eSCI_CR2[FEIE]       |
| Receiver              | SCI              | eSCI_IFSR1[PF]    | eSCI_CR2[PFIE]       |
| Receiver              | LIN              | eSCI_IFSR1[BERR]  | eSCI_CR2[BERRIE]     |
| Receiver              | LIN              | eSCI_IFSR2[RXRDY] | eSCI_LCR1[RXIE]      |
| Transmitter           | LIN              | eSCI_IFSR2[TXRDY] | eSCI_LCR1[TXIE]      |
| Receiver              | LIN              | eSCI_IFSR2[LWAKE] | eSCI_LCR1[WUIE]      |
| Receiver              | LIN              | eSCI_IFSR2[STO]   | eSCI_LCR1[STIE]      |
| Receiver              | LIN              | eSCI_IFSR2[PBERR] | eSCI_LCR1[PBIE]      |
| Receiver              | LIN              | eSCI_IFSR2[CEERR] | eSCI_LCR1[CIE]       |
| Receiver              | LIN              | eSCI_IFSR2[CKERR] | eSCI_LCR1[CKIE]      |
| Receiver              | LIN              | eSCI_IFSR2[FRC]   | eSCI_LCR1[FCIE]      |
| Receiver              | LIN              | eSCI_IFSR2[UREQ]  | eSCI_LCR2[URIE]      |
| Transmitter, Receiver | LIN              | eSCI_IFSR2[OVFL]  | eSCI_LCR2[OFIE]      |



### 42.5.7.2 Interrupt Request Generation

The eSCI module provides one hardware interrupt request signal to the systems interrupt controller. This interrupt request signal is asserted if and only if at least one of the interrupt flags and the corresponding interrupt enables are set to 1. Otherwise the interrupt line is deasserted.

## 42.6 Application Information

### 42.6.1 SCI Data Frames Separated by Preamble

To separate SCI data frame with preambles with minimum idle line time, use this sequence between messages:

1. write to SCI Data Register (eSCI\_SDR)
  - this sets the internal iCMT bit which requests the data transmission
2. wait until TDRE in Interrupt Flag and Status Register 1 (eSCI\_IFSR1) is set
  - this indicates the start of transmission; the iCMT bit was cleared
3. clear and subsequently set the TE bit in Control Register 1 (eSCI\_CR1)
  - this sets the internal iPRE bit which requests the preamble transmission
4. write to SCI Data Register (eSCI\_SDR)
  - this sets the internal iCMT bit which requests the data transmission

The priority scheme of the transmitter which is described in [Table 42-29](#) ensures that the preamble is transmitted before the data frame.



# Chapter 43

## Resets

### 43.1 Overview

This device supports the following system reset sources:

- Power-on reset (at power up)
- External reset (from RESET pin)
- Fault Collection and Control Unit (FCCU) reset
- FCCU Output Supervision Unit (FOSU) reset
- Dual core reset
- JTAG reset
- Self Test Control Unit (STCU) reset
- Software system reset
- Life cycle manipulation reset

A reset event caused by any of the above sources will trigger a reset cycle with the assertion of internal reset signals as well as the assertion of the RSTOUT pin. Software may set the SIU\_SRCR[SER] bit to 1 to assert the RSTOUT pin without causing an internal reset cycle. This action is referred to as a software external reset. After the reset cycle completes, software may learn the source of the last reset cycle by looking at the Reset Status Register (SIU\_RSR) in the SIU. The SIU\_RSR register is updated for all reset sources except JTAG and life cycle manipulation reset.

Power-on reset is the most comprehensive form of reset. It affects all modules without exception. Other sources of reset do not affect special hardware that needs to be active during the reset cycle, like for example the state machine of the reset controller. Some modules are affected by all reset sources, but are released from reset state before other modules during the execution of the reset cycle. This reset releasing sequence allows device configuration data to be read from flash memory and applied to the system before the completion of the reset cycle and prior the cores starting code execution.

When the reset cycle completes, the RSTOUT pin is negated. In parallel, the CSE security engine starts execution of the secure boot, and Core 0 starts execution. Core 1 and the associated checker core are kept in reset state until software running on Core 0 configures its reset vector and enables it to start code execution. Reset vectors and core resets are configured by the SIU\_RSTVEC0 and SIU\_RSTVEC1 registers.

The reset cycle can be extended by the assertion of the RESET pin. This is useful to allow JTAG communication to be established during the reset cycle, so that debugging configuration may be done before the cores start executing.

## 43.2 Reset sequence

The following figure illustrates the reset flow.

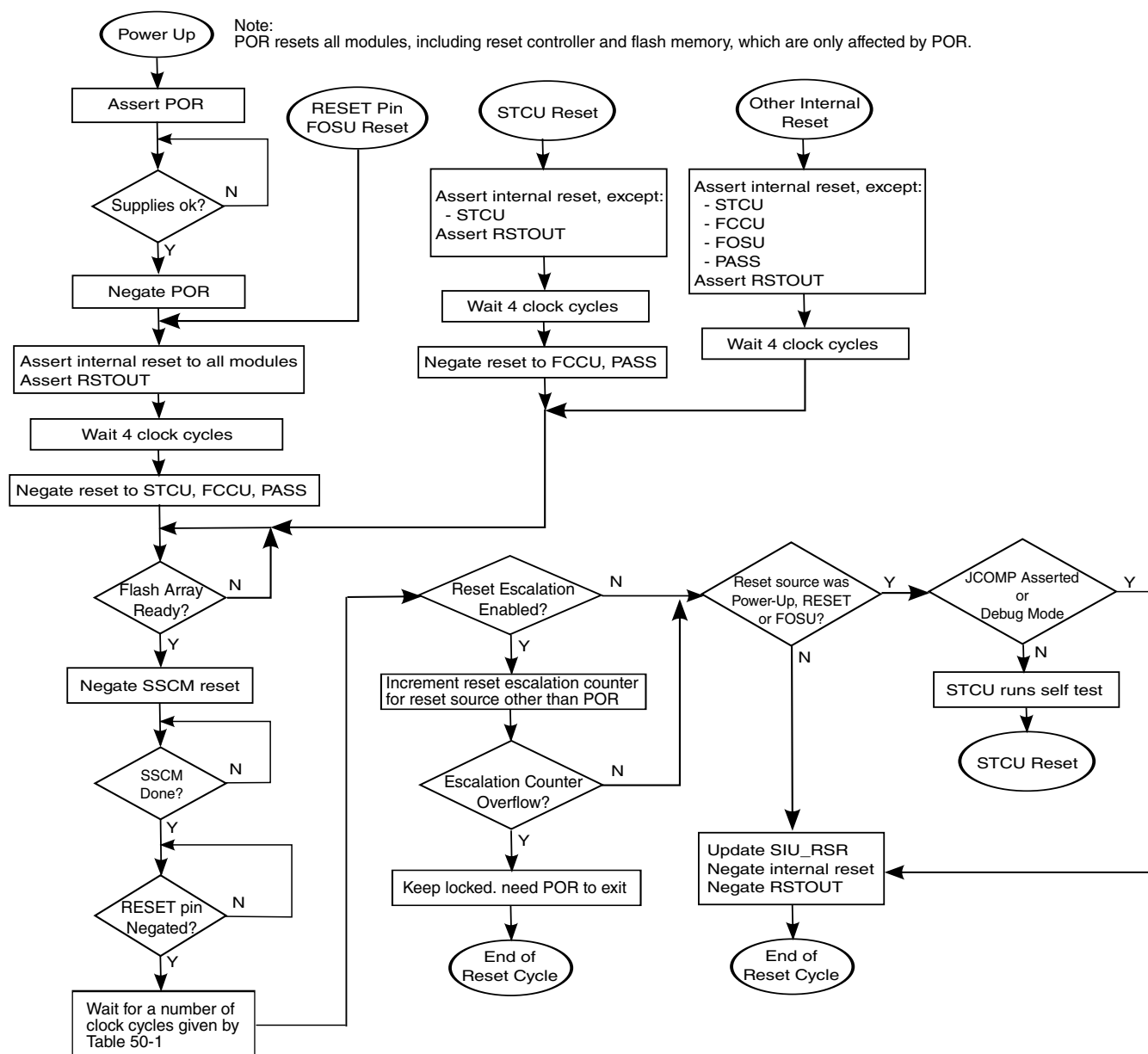


Figure 43-1. Internal reset flow diagram

When power-up is the source of the reset cycle, the following sequence of events takes place:

1. A power-on reset (POR) signal resets everything while power supplies are ramping up.
2. When supply rails are within acceptable (yet still untrimmed) voltage ranges, the POR signal is negated, the RSTOUT pin is asserted, and an internal reset signal is asserted to all modules.

**NOTE**

Some components are exceptions to this internal reset assertion. They include the reset controller, flash memory, and DCF clients (ERROR function and Reset escalation), all of which are affected only by POR.

3. The reset controller waits for 4 clock cycles so that the reset signal can propagate through the registers, then it negates the reset signals going to the STCU, FCCU and PASS, because these modules need to be active during the reset cycle.
4. The reset controller waits until the flash memory finishes its initialization, then negates the reset signal going to the SSCM module.
5. The SSCM reads device configuration and trimming data from DCF records stored in flash memory and applies the configuration data to the system. Refer to the DCF chapter for more information.
6. When the SSCM is done, the reset controller waits until the RESET pin is negated and then starts a timer that counts for a pre-determined number of clock cycles, which depends on the reset source (see [RSTOUT](#)).
7. If reset escalation is enabled, the reset cycle proceeds with reset escalation evaluation. A reset escalation counter is incremented at the end of every non-POR reset cycle. If the counter exceeds the number programmed in the RET field of SIU\_RCR, then the device is locked permanently into a reset state which can only be undone by a power-up cycle. To avoid reaching the reset escalation threshold, software must reset the escalation counter by servicing the SIU\_RCR register (write something to the RET field; writing zero disables the reset escalation process).
8. If the device was not locked at reset state by the reset escalation process, the STCU may take control of clocks and resets of the device to execute self-test diagnostics. Self-test is executed during the reset cycle when the following conditions are satisfied:
  - The reset cycle was initiated from power-up or RESET pin or FOSU reset
  - STCU execution is enabled by DCF configuration
  - JCOMP pin is not asserted and debug mode is not engaged

After the self-test is finished, the STCU records the test results and then triggers a new internal reset cycle via an STCU reset request. This new reset cycle will not trigger a self-test execution.

9. If the STCU did not execute, then the reset cycle reaches its normal termination flow, the SIU\_RSR register is updated, the RSTOUT pin is negated, and the internal reset signals are negated.

Reset cycles initiated by sources other than power-up, RESET pin, or FOSU do not affect all modules. For a reset cycle initiated by the STCU, the STCU itself is not reset, so that diagnostic results from the previous reset cycle are preserved.<sup>1</sup> Other sources of internal reset will not reset the STCU, FCCU, FOSU and PASS modules (they have information that needs to be preserved through the reset cycle).

## 43.3 Reset pins

### 43.3.1 RESET

The RESET pin is an active low input. The RESET pin must be asserted by an external device during power-up or whenever an external reset is required.

The RESET pin has hysteresis, and the hysteresis is always enabled.

The reset controller provides a glitch detector connected to the RESET pin. If the reset controller detects that the RESET pin is asserted for more than two peripheral clock cycles, the event is latched. After the latch is set, if the RESET pin is negated before 10 clock cycles is reached, the reset controller sets the RGF bit without affecting any of the other bits in the reset status register (SIU\_RSR). The latch is cleared when the RGF bit is set or a valid reset is recognized. The RGF bit remains set until cleared by a software or the RESET signal asserts for 10 clock cycles.

### 43.3.2 RSTOUT

The RSTOUT pin is an active low output that uses a push/pull configuration. The RSTOUT pin is driven low by the MCU for all internal and external reset sources.

The time interval between RESET pin negation and RSTOUT pin negation is deterministic, but it depends on the reset source, as shown in the following table. The actual time interval is the value in [Table 43-1](#) plus 4 clock cycles.

The RSTOUT pin can also be asserted by a write to the SER bit of the System Reset Control Register (SIU\_SRCR); however, an internal reset cycle is not initiated under this circumstance. The duration of RSTOUT assertion caused by setting the SER bit depends on whether the clock source is the IRC or not, as given in the following table.

---

1. The FCCU needs to be reset after a self-test run, because FCCU logic is part of the LBIST tests.

**Table 43-1. Clock Cycles for Different Reset Sources**

| Source <sup>1</sup> | Description                          | Number of clock cycles     |
|---------------------|--------------------------------------|----------------------------|
| PORS                | Power-on reset                       | 2400                       |
| ERS                 | External reset (RESET pin)           | 2410                       |
| FOSURS              | FOSU reset                           | 2430                       |
| FCCURS              | FCCU reset                           | 2460                       |
| CPURS               | Simultaneous core 0 and core 1 reset | 2490                       |
| STCURS              | STCU reset                           | 2500                       |
| SSRS                | Software system reset                | 2510                       |
| SERF <sup>2</sup>   | Software external reset              | 2520 (IRC) / 17200 (other) |

1. Bits of the SIU\_RSR register.

2. RSTOUT duration depends on the clock source: if IRC, duration is 2520; for all others, duration is 17200.

## 43.4 Reset sources

### 43.4.1 Power-on reset

The internal power-on reset signal is asserted when voltage and temperature monitors indicate that the device temperature and/or power supply voltages have reached values outside normal operating ranges. Control and configuration registers for temperature and voltage monitors are located within the Power Management Control (PMC) digital interface module.

POR is the highest priority reset and if POR is detected while any other reset is being processed, after reset cycle completion SRS will have just the PORS set.

#### 43.4.1.1 Voltage monitors

The following out-of-range voltage events can trigger a power-up reset cycle:

- POR: below absolute minimum value required for starting device wake-up process
- Low voltage Detection (LVD): below minimum power supply voltage specification
- High Voltage Detection (HVD): above maximum power supply voltage specification



A power-up reset cycle is triggered unconditionally when supply voltages fall below POR thresholds. LVD and HVD events can also trigger a power-up cycle if enabled in the PMC module. LVD/HVD events are also routed from the PMC to the FCCU to be handled there.

LVD/HVD events which are enabled to trigger reset will cause a POR reset cycle. When the reset cycle finishes, the PORS flag is set in the SIU\_RSR register. If LVD/HVD are routed to the FCCU, and the FCCU is enabled to generate reset from these sources, then an FCCU reset cycle is triggered. When the FCCU reset cycle finishes, the FCCURS bit is set in the SIU\_RSR register. The FCCU reset cycle will not reset the FCCU itself, therefore information about the LVD/HVD events can still be recovered from the FCCU registers.

See the the PMC chapter and the device data sheet for information on monitored voltages and threshold values.

### 43.4.1.2 Temperature monitors

The following table identifies the temperatures for which out-of-range temperature events can trigger a power-up reset cycle:

**Table 43-2. Temperature events configurable to cause POR**

| Temperature | Temperature Sensor name | PMC name      | Out-of-range event                                       |
|-------------|-------------------------|---------------|----------------------------------------------------------|
| -40°C       | Low temperature         | Temperature 0 | Temperature is below this threshold                      |
| 150°C       | High temperature 1      | Temperature 2 | Temperature is above this threshold                      |
| 165°C       | High temperature 2      | n/a           | No event: This device does not support 165°C monitoring. |

These events trigger a power-up cycle if enabled in the PMC module. Alternatively, the PMC can be configured to route the temperature events to the FCCU to be handled there.

Temperature events that are enabled to trigger a reset cause a POR cycle. When the reset cycle finishes, the PORS flag is set in the SIU\_RSR register. If the temperature events are routed to the FCCU, and the FCCU is enabled to generate reset from these sources, then an FCCU reset cycle is triggered. When the FCCU reset cycle finishes, the FCCURS bit is set in the SIU\_RSR register. The FCCU reset cycle will not reset the FCCU itself, so information about the LVD/HVD events can still be recovered from the FCCU registers.

See the detailed descriptions of the Temperature Sensor and PMC modules as well as the device Data Sheet for information about monitored temperatures and threshold values.

### 43.4.2 External reset

Assertion of the RESET pin triggers an internal reset cycle. The reset cycle will be sustained as long as RESET is kept asserted. When RESET is negated, the reset controller initiates a timer that counts for a predetermined number of clock cycles, as shown in [Table 43-1](#), before resuming the reset cycle.

If the RESET pin was the source reset, or if it was asserted during the last reset cycle, then the ERS bit is asserted in the SIU\_RSR register.

If POR is the current reset source being processed, then a RESET pin is detected, after reset cycle completion SRS will have PORS and ERS set.

If the Reset pin is the current reset source being processed, then if any other reset is detected (except POR), after reset cycle completion SRS will have just ERS set and PORS will keep its previous value.

### 43.4.3 FCCU reset

The Fault Collection and Control Unit collects failure events coming from different sub-systems within the device. One of the possible reactions to a failure event is the triggering of an internal reset cycle.

Failure sources include:

- All failure indication signals from modules within the MCU
- Logic and signals monitored by the FCCU itself
- Software-initiated failure indications
- External failure input.

Available failure reactions are:

- Assertion of an interrupt (maskable or nonmaskable)
- Resetting the MCU
- Changing the state of the failure indication pins, ERRORn
- No reaction

The FCCU itself is not affected by an FCCU reset. The FCCU is reset when the reset source is one of the following:

- Power-up
- RESET pin
- FOSU reset
- STCU reset

When the reset cycle that was initiated by the FCCU finishes, the FCCURS bit is asserted in the SIU\_RSR register. Since the FCCU itself is not reset when the reset source is an FCCU reset, information about the failure that caused reset is still available in the FCCU module to be read by software.

Refer to the FCCU chapter for information on failure sources that can generate a reset cycle.

#### 43.4.4 FOSU reset

As the FCCU is a central component in reacting to errors, it is itself supervised. This supervision is provided by the FCCU Output Supervision Unit (FOSU). The FOSU receives failure indications at the same time as the FCCU. Unless the respective failure is switched off, the FOSU will observe the outputs of the FCCU (IRQ, RESET, error pins). If the FCCU does not react within a predefined timing window, the FOSU assumes the FCCU has failed and causes a reset cycle.

A FOSU reset cycle resets the FCCU and the FOSU itself. The FOSU is reset when the reset source is one of the following:

- Power-up
- RESET pin
- FOSU reset
- STCU reset

When the FOSU reset cycle finishes, the FOSURS bit is asserted in the SIU\_RSR register.

#### 43.4.5 Dual core reset

Core 0 and Core 1 Reset Vector Registers described in the SIU chapter allow software executing on either core to place the other core in reset by setting the RST bit in the appropriate Reset Vector Register. Any attempt to reset both cores by setting both RST bits will result in an immediate internal reset.

When the reset cycle that was initiated by this reset source finishes, the CPURS bit is asserted in the SIU\_RSR register.

#### 43.4.6 JTAG reset

An internal reset cycle is initiated when JTAG is enabled and either the EXTEST, CLAMP, or HIGHZ instructions are executed by the JTAG controller. Since these instructions affect the state of the package pins, the state of the RSTOUT pin is determined by the JTAG instruction itself. The reset status bits in the SIU\_RSR are not affected by this reset source.

### **43.4.7 Software system reset**

A software system reset occurs when software asserts the SSR bit in the System Reset Control Register (SIU\_SRCR), triggering an internal reset cycle and the assertion of the RSTOUT pin. Once the reset cycle begins, the SSR bit is automatically cleared. When the reset cycle finishes, the SSRS bit is set in the SIU\_RSR register.

### **43.4.8 Software external reset**

A write of one to the SER bit in the SIU\_SRCR causes the external RSTOUT pin to be asserted for a predetermined number of clock cycles (refer to [RSTOUT](#)). The SER bit automatically clears after the clock counting expires. A software external reset does not cause an internal reset cycle. Only the RSTOUT pin is affected. The SERF bit in the SIU\_RSR is set, but no other status bits are affected. The SERF bit in the SIU\_RSR is not automatically cleared after the RSTOUT clock cycles expire, and remains set until cleared by software or another reset besides the software external reset occurs.

During a software external reset, the e200z7 cores will continue to execute instructions, timers that are enabled will continue to operate, and interrupt requests will continue to be processed. It is the responsibility of the application to ensure devices connected to RSTOUT are not accessed during a software external reset, and to determine how to manage MCU resources when using the software external reset.

### **43.4.9 STCU reset**

When the reset cycle is triggered by a power-up event or by the assertion of the RESET pin or a FOSU reset, and if enabled by DCF configuration, the Self-Test Control Unit executes a series of hardware self tests based on shifting pseudo-random data into the device registers and then comparing the resulting signature with a golden reference stored in flash memory. It also triggers the execution of self-test hardware within the embedded

memories. By the end of the self-test runs, the device is left with random data in its registers; therefore, the STCU sends a signal to the reset controller to trigger a new reset cycle.

The STCU itself is not affected by STCU reset cycle; therefore, test results are preserved to be read by software after the reset cycle finishes. The FCCU is also one of the structures included in the STCU tests; therefore, it also needs to be reset after the execution of STCU diagnostics.

When the reset cycle that was initiated by this reset source finishes, the STCURS bit is asserted in the SIU\_RSR register.

### 43.4.10 Life cycle manipulation reset

When SSCM loads information from UTest memory corresponding to the life cycle slots and that information includes invalid programmed values, a life cycle manipulation reset occurs. See [SSCM security reset and LC state](#) for further details.

## 43.5 Reset configuration

During reset, the SSCM reads various configuration and trimming data from flash memory and applies them to the device before the reset cycle finishes. See the DCF chapter for a complete list of DCF records. Following is a list of main functions configured by DCF records and information in other locations in UTest flash memory:

- Pull up/down of external I/O pins
- Boot mode (internal, serial, EBI, auto-baud)
- Crystal oscillator trimming and configuration
- RC oscillator trimming
- Trimming for voltage regulators and power supply monitors
- ADC trimming
- Temperature sensor trimming
- Flash access control bits and passwords
- Security, life cycle, and censorship
- STCU configuration

### 43.5.1 Weak pull up/down configuration

Initial pull up/down configuration of external I/O pins is available in the Miscellaneous DCF record, which is read by the SSCM from UTEST flash memory during reset and applied to the pad drivers.

As an alternative to the DCF record method, a legacy mode is available that uses the WKPCFG pin to do the same configuration. For all resets except a software external reset, the WKPCFG pin is applied at the assertion of the internal reset signal and latched upon negation of reset. Legacy mode is used when the legacy bit is asserted in the Miscellaneous DCF record.

Either coming from the DCF record or from the WKPCFG pin, the pull configuration is stored in the WKPCFG bit in the Reset Status Register.

### 43.5.2 Boot configuration

Configuration data required by the boot code is available in the Miscellaneous DCF record, which is read by the SSCM from UTEST flash memory during the reset cycle. As an alternative to the DCF record method, a legacy mode is available in which the boot configuration data is latched from package pins at the end of the reset cycle. Either coming from the DCF record or from package pins, the boot configuration data is placed in the Reset Status Register (SIU\_RSR) to be read by the boot code.

The boot configuration data in SIU\_RSR is also affected by the security life cycle of the component. If the life cycle has reached the “in field” state, then boot mode is forced to be internal only. See the table below and also refer to the SIU\_RSR description in the SIU chapter.

**Table 43-3. Boot configuration data**

| Mode (LEG bit in Misc DCF record) | SIU_RSR fields                     |                                    |
|-----------------------------------|------------------------------------|------------------------------------|
|                                   | BOOTCFG <sup>1</sup>               | ABR                                |
| Normal (LEG=0)                    | Loaded from bits 14–15 of Misc DCF | Loaded from bit 13 of Misc DCF     |
| Legacy (LEG=1)                    | Latched from BOOTCFG[0:1] pins     | Latched from EVTO pin <sup>2</sup> |

1. BOOTCFG is forced to 0b00 (internal boot) if life cycle is “in-field.”

2. Latching from EVTO works as follows. If no debug tool is attached to the device and EVTO is held at low logic level during reset, the ABR bit reset state is logic one. Otherwise it is zero.

## 43.6 Boot sequence

When the reset cycle finishes, the following sequence of events takes place:

1. In parallel, immediately after reset is negated, the Cryptographic Services Engine (CSE) starts running its secure boot code, and core 0 starts executing boot code from address 0xFFFF\_FFFC.
3. Core 1 is held in reset state until software writes a boot vector in SIU\_RSTVEC1 and then negates SIU\_HLT1[CORE1] or SIU\_RSTVEC1[RST1]. When either of these bits is negated, core 1 starts executing code at the address specified by the boot vector.
4. Execution of the checker core is controlled by the LOCKSTEP\_EN bit of the Miscellaneous DCF Client. If disabled, the checker core is kept in reset state after the end of the reset cycle even if core 1 starts executing code. When the checker core is disabled, RCCU error checking is also disabled. Enabling or disabling checker core execution requires a reset cycle to take effect.





# Chapter 44

## System Boot

### 44.1 Overview

The Boot Assist Module (BAM) is a 4-KB block of read-only memory, containing the boot program code for the MCU. The BAM program supports the following boot modes:

- Boot from internal flash memory
- Serial boot via SCI or CAN interface with optional baud-rate detection (available if the device life cycle is not “in-field”)
- Boot from a memory connected to the MCU development bus (EBI)<sup>1</sup> with multiplexed or separate address and data lines (available if the device life cycle is not “in-field”)

On the MPC5775E device, only Core0 is enabled out of reset; therefore, the BAM program is executed by Core0 after the reset cycle finishes. Depending on the boot mode, the program initializes the appropriate minimum MCU resources to start user application code. Note that the BAM is pure software; therefore, all registers and resources accessed and modified by the BAM can also be accessed by customer software.

### 44.2 Features

- Initial MMU setup with minimum address translation for all internal MCU resources
- MMU configuration to boot user application, compiled as Classic PowerPC Book E code or as NXP VLE code
- Passes control to user application code in the internal flash memory or external memory device
- Automatic switch to serial boot mode if internal or external flash is blank or invalid
- Serial boot by loading user program via CAN bus or eSCI to the internal SRAM
  - Not available at “in-field” device life cycle

---

1. EBI not available on all packages.

- Only public password protection is used
- Optional automatic detection of the host SCI or CAN speed
- Boot from an external memory device, connected to the EBI
  - Not available at “in-field” device life cycle
  - Option to boot from 16-bit memory device with separate data and address lines
  - Option to boot from 32/16-bit memory device with multiplexed data and address lines
- Controls MCU core Watchdog Timer and/or the Software Watchdog Timer (SWT)

## 44.3 Modes of operation

### 44.3.1 Normal boot

Core 0 is released from reset immediately at the end of the reset cycle and it begins execution of the BAM code from the reset vector contained in the SIU\_RSTVEC0 register.

### 44.3.2 Parallel secure boot

The Cryptographic Service Engine (CSE) and Core0 are released from reset at the same time. Core0 starts executing the BAM code in parallel with CSE execution. The BAM code is ignorant of the result of the secure boot code execution; it always continues with its own code execution.

### 44.3.3 Debug mode

The BAM program is not executed when the MCU exits reset in OnCE debug mode. Before accessing the MCU resources, use the development tool to initialize the MCU.

### 44.3.4 Internal boot mode

BAM looks for customer code at predefined locations in flash memory, then starts execution of that code from flash.

### 44.3.5 Serial boot mode

BAM loads customer code into internal SRAM using the eSCI or CAN serial interface, then executes that program. The program can then control:

- Downloading of data, and
- Erasing and programming of the internal or external flash memory

### 44.3.6 Development bus boot mode

This boot mode is intended for packaged versions of the device that have the development bus pinned out. The development bus (EBI) boot mode can be used for application code development.

## 44.4 Memory Map

The BAM occupies 16 KB of memory space, 0xFFFF\_C000 to 0xFFFF\_FFFF. The actual code size of the BAM program is less than 4 KB and starts at 0xFFFF\_F000, repeating itself every 4 KB in the BAM address space. The CPU starts the BAM program execution at its reset vector from address 0xFFFF\_FFFC.

The BAM exits to user code at 0xFFFF\_FFF8. The last instruction the BAM executes is a BLR. The link register is pre-loaded with the user application start address. The value of the start address depends on the boot mode:

- Booting from internal or external flash memory—32-bit word following a valid RCHW holds the start address value
- Serial boot—set according to the serial boot protocol

The following table shows the BAM address map.

**Table 44-1. BAM memory map**

| Address                   | Description                   |
|---------------------------|-------------------------------|
| 0xFFFF_C000 – 0xFFFF_EFFF | BAM program mirrored          |
| 0xFFFF_F000 – 0xFFFF_FFFF | BAM program                   |
| 0xFFFF_FFFC               | MCU reset vector              |
| 0xFFFF_FFF8               | BAM last executed instruction |

## 44.5 Functional Description

### 44.5.1 BAM program flow chart

The following figure shows the BAM program flow.

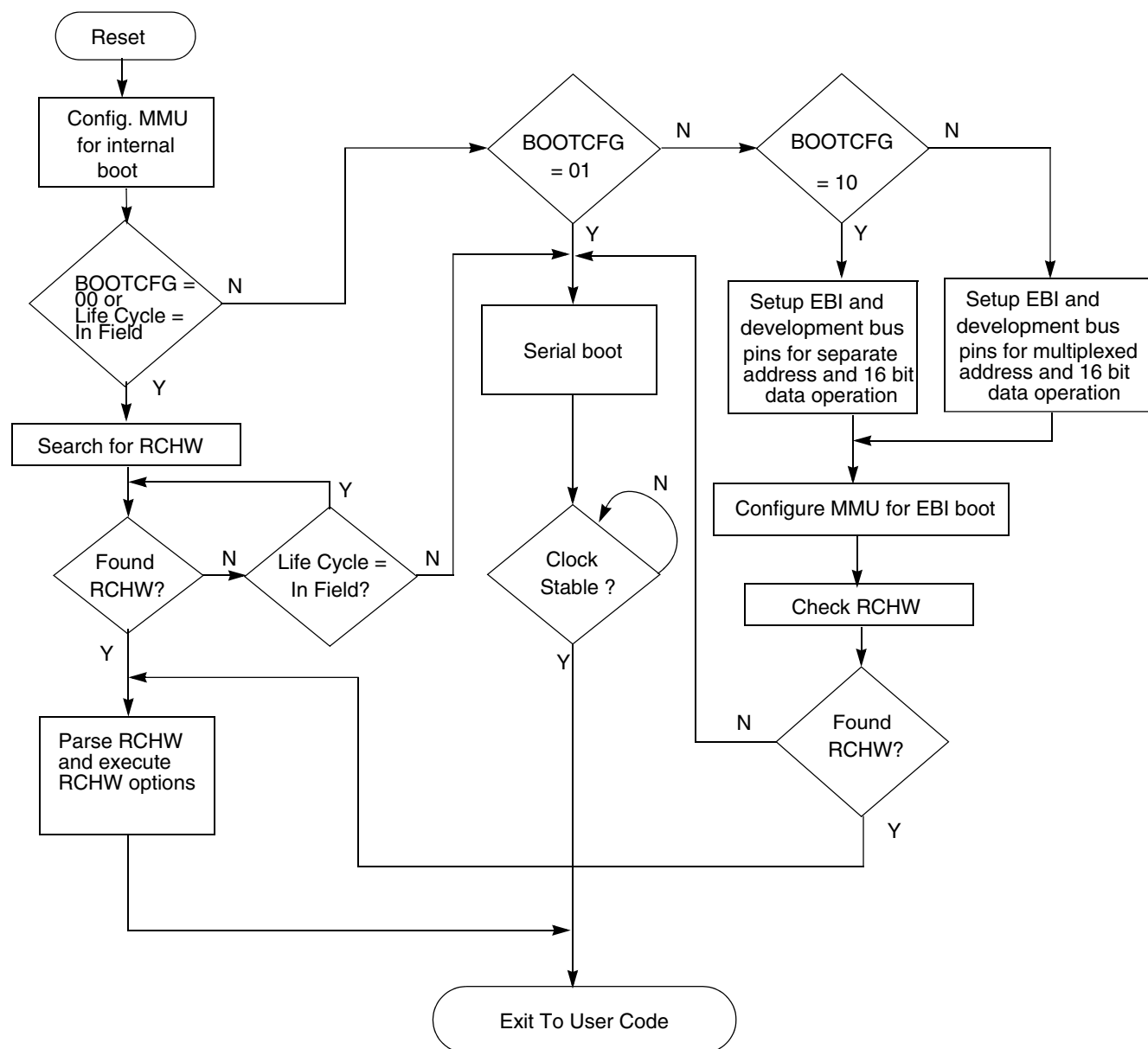


Figure 44-1. BAM program flow chart

## 44.5.2 BAM program operation

When Core0 is released from reset state, it fetches the boot vector corresponding to the BAM code from the SIU\_RSTVEC0 register. The first thing the BAM program does is to configure the MMU to allow access to all MCU internal resources, according to [Table 44-2](#). This MMU setup remains untouched through internal flash boot mode.

**Table 44-2. MMU configuration for internal flash boot**

| TLB entry | Region                      | Logical base address | Physical base address | Size   | Attributes    |
|-----------|-----------------------------|----------------------|-----------------------|--------|---------------|
| 0         | Peripheral Bridge B and BAM | 0xFFE0_0000          | 0xFFE0_0000           | 2 MB   | Guarded       |
|           |                             |                      |                       |        | Big endian    |
|           |                             |                      |                       |        | Global PID    |
|           |                             |                      |                       |        | Cache Inhibit |
| 1         | Internal flash              | 0x0000_0000          | 0x0000_0000           | 16 MB  | Not guarded   |
|           |                             |                      |                       |        | Big endian    |
|           |                             |                      |                       |        | Global PID    |
| 2         | EBI                         | 0x2000_0000          | 0x0000_0000           | 16 MB  | Not guarded   |
|           |                             |                      |                       |        | Big endian    |
|           |                             |                      |                       |        | Global PID    |
| 3         | Internal SRAM               | 0x4000_0000          | 0x4000_0000           | 512 KB | Not guarded   |
|           |                             |                      |                       |        | Big endian    |
|           |                             |                      |                       |        | Global PID    |
|           |                             |                      |                       |        | Cache Inhibit |
| 4         | Peripheral Bridge A         | 0xC3E0_0000          | 0xC3E0_0000           | 2 MB   | Guarded       |
|           |                             |                      |                       |        | Big endian    |
|           |                             |                      |                       |        | Global PID    |
|           |                             |                      |                       |        | Cache Inhibit |

The MMU regions are mapped with logical addresses the same as physical addresses except for the External Bus Interface (EBI). The logical EBI address space is mapped to physical address space of the internal flash memory. This allows code, written to run from external memory, to execute from internal flash memory.

After the MMU configuration, the BAM uses information from the read-only SIU\_RSR[BOOTCFG] field to decide what type of boot that it must do, according to [Table 44-3](#).

**Table 44-3. BOOTCFG field description**

| BOOTCFG[0:1] | Description                      |
|--------------|----------------------------------|
| 00           | Internal boot from flash memory  |
| 01           | Serial boot from eSCI or FlexCAN |

*Table continues on the next page...*

**Table 44-3. BOOTCFG field description (continued)**

| BOOTCFG[0:1] | Description                                             |
|--------------|---------------------------------------------------------|
| 10           | Non-muxed EBI boot (16-bit data, non-muxed address bus) |
| 11           | Muxed EBI boot mode (16-bit data, muxed address bus)    |

The BOOTCFG field is the only place that the BAM will look to determine what type of boot it should attempt. The value of the BOOTCFG field is loaded into the SIU\_RSR by internal hardware during reset, from one of the following possible sources:<sup>2</sup>

1. BOOTCFG pins
2. BOOTCFG field of the Miscellaneous DCF record
3. Fixed value 00 (Internal Boot)

Which of these sources is used depends on the life cycle and the legacy mode settings of the device.

Table 44-4 describes the source selection of the BOOTCFG field depending on the above settings.

**Table 44-4. BOOTCFG source selection**

| Life cycle              | Legacy mode?   | BOOTCFG source                            |
|-------------------------|----------------|-------------------------------------------|
| Different than In Field | Yes            | BOOTCFG pins                              |
|                         | No             | BOOTCFG field of Miscellaneous DCF record |
| In Field                | X <sup>1</sup> | Fixed to 00 (Internal Boot)               |

1. "X" means that the value of this entry has no effect on the determination of the BOOTCFG source.

For all cases presented where serial boot mode is available, a 64-bit public password is required. The password is 0xFEED\_FACE\_CAFE\_BEEF. After a match is detected, the BAM code goes on executing the serial boot flow. Otherwise, it stays on a loop until a reset is issued due to the watchdog settings.

The internal and EBI boot modes require no password, but it is still possible to have the flash memory inaccessible due to protection settings on the PASS module. Therefore, access to flash should be granted by not blocking the flash memory regions used by the BAM.

2. The BOOTCFG field is the only place that the BAM will look to determine what type of boot it should attempt. Other device settings affect the boot mode, but they are all factored in by the internal hardware when it loads BOOTCFG into the SIU\_RSR register.

### 44.5.3 Reset Configuration Half Word (RCHW)

The Reset Configuration Half Word defines boot options and must be programmed to predefined locations in the internal flash or at the beginning of the external flash device. The 32-bit word after the RCHW must be programmed with the user application's starting address. The BAM passes control to the user application at this starting address.

Table 44-6 provides possible RCHW locations in the internal flash. When booting from the external flash device, the RCHW must reside in the first 16-bit half word of the flash memory.

BOOT\_BLOCK\_ADDRESS

| 0 | 1 | 2 | 3 | 4   | 5   | 6   | 7   | 8                      | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|-----|-----|-----|-----|------------------------|---|----|----|----|----|----|----|
|   |   |   |   | SWT | WTE | PS0 | VLE | 0                      | 1 | 0  | 1  | 1  | 0  | 1  | 0  |
|   |   |   |   |     |     |     |     | Boot Identifier = 0x5A |   |    |    |    |    |    |    |

**Table 44-5. RCHW field descriptions**

| Field               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–3                 | Reserved. These bit values are ignored when the halfword is read. Program to 0 for future compatibility.                                                                                                                                                                                                                                                                                                                                                                                                         |
| 4<br>SWT            | Software watchdog timer enable. This bit determines if the software watchdog timer is enabled after passing control to the user application code.<br>0 Disable software watchdog timer<br>1 Enable software watchdog timer after reset. The time-out period is 392400 cycles of the Internal RC Oscillator.                                                                                                                                                                                                      |
| 5<br>WTE            | MCU core watchdog timer enable. This bit determines if the core software watchdog timer is enabled after passing control to the user application code.<br>0 Disable core software watchdog timer<br>1 Enable core watchdog timer after reset. The timeout period is $2.5 \times 2^{18}$ cycles of the CPU clock. <sup>1</sup>                                                                                                                                                                                    |
| 6<br>PS0            | Port size. Defines the width of the data bus connected to the memory on D_CS0. After system reset, the BAM changes D_CS0 to a 16-bit port to fetch the RCHW from either 16- or 32-bit external memories. Then the BAM reconfigures the EBI as a 16- or 32-bit port, depending on this bit.<br>0 32-bit D_CS0 port size<br>1 16-bit D_CS0 port size<br><b>NOTE:</b> Used in development bus boot modes only (that is, 496- and 516-pin packages). Do not clear this bit if the device only has a 16-bit data bus. |
| 7<br>VLE            | VLE Code Indicator. This bit configures the MMU entries 1–3 coded as Classic Book E instructions or as NXP VLE instructions.<br>0 User code executes as classic Book E code<br>1 User code executes as NXP VLE code                                                                                                                                                                                                                                                                                              |
| 8–15<br>BOOTID[0:7] | Boot identifier. This field serves two functions: <ul style="list-style-type: none"> <li>Indicates which block in flash memory contains the boot program</li> <li>Indicates if the flash memory is programmed (BOOTID = 0x5A) or invalid</li> </ul>                                                                                                                                                                                                                                                              |

1. Upon exiting reset, the CPU clock is connected to the Internal RC Oscillator at 16 MHz.

### 44.5.3.1 Application Start Address Register

Application Start Address Register, shown below, is the 32-bit word after the RCHW in the application code memory device (internal or external flash memory). The BAM uses the value from this location as a start address of the user application to pass control to.

BOOT\_BLOCK\_ADDRESS + 0x0000\_0004

|    |    |    |    |    |    |    |    |    |    |     |     |     |     |     |     |
|----|----|----|----|----|----|----|----|----|----|-----|-----|-----|-----|-----|-----|
| 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10  | 11  | 12  | 13  | 14  | 15  |
| A0 | A1 | A2 | A3 | A4 | A5 | A6 | A7 | A8 | A9 | A10 | A11 | A12 | A13 | A14 | A15 |

|     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 16  | 17  | 18  | 19  | 20  | 21  | 22  | 23  | 24  | 25  | 26  | 27  | 28  | 29  | 30  | 31  |
| A16 | A17 | A18 | A19 | A20 | A21 | A22 | A23 | A24 | A25 | A26 | A27 | A28 | A29 | A30 | A31 |

### 44.5.4 Internal boot mode

Upon reset exit, when the BAM program starts running, the system clock is by default sourced by the Internal RC Oscillator (IRC). If internal boot mode was configured in the SIU\_RSR[BOOTCFG] field, the BAM program executes the following tasks:

1. Configures a machine check exception handler, since it will be accessing flash memory locations that may be corrupted and cause a bus error
2. Searches for a valid RCHW in predefined locations, as shown in the following table.

#### NOTE

In the search for a valid RCHW, the location with priority 0 has highest priority and the location with priority 4 has lowest priority.

**Table 44-6. Possible RCHW locations in internal flash memory**

| Priority | Address     |
|----------|-------------|
| 0        | 0x0080_0000 |
| 1        | 0x0002_0000 |
| 2        | 0x0003_0000 |
| 3        | 0x0000_0000 |
| 4        | 0x0001_0000 |

3. If a valid RCHW is not found, the BAM program checks for the life cycle state of the device (SSCM\_LCSTAT[LC]). For any life cycle other than In Field, the BAM program switches to the serial boot mode. Otherwise, the search for the RCHW will continue until SWT requests a reset.
4. If the BAM program finds a valid RCHW:



- `BOOT_BLOCK_ADDRESS` is the address (from [Table 44-6](#)) of the valid RCHW
- The BAM program fetches the application start address from `BOOT_BLOCK_ADDRESS + 0x4`
- The BAM branches to this start address (shown in [Application Start Address Register](#))
- The MMU TLB entries 1,2,3 are programmed with VLE attribute if the `RCHW[VLE] = 1`
- The watchdog timers are programmed as shown in the table below

**Table 44-7. Watchdog timeout settings**

| Core WD       |                                                                               | SWT           |                                                                  |
|---------------|-------------------------------------------------------------------------------|---------------|------------------------------------------------------------------|
| Clock source  | Condition                                                                     | Clock source  | Condition                                                        |
| IRC<br>16 MHz | Enabled if <code>RCHW[WTE] = 1</code><br>Timeout = $2.5 \times 2^{18}$ cycles | IRC<br>16 MHz | Enabled if <code>RCHW[SWT] = 1</code><br>Timeout = 392400 cycles |

### 44.5.5 Serial boot mode

When serial boot mode is detected, either because of the `BOOTCFG` configuration or because a valid RCHW word was not found in any of the expected locations in flash memory, the BAM program does the following:

1. Initializes the FlexCAN\_A RAM to avoid ECC errors
2. Uses message buffers in FlexCAN\_A for program stack and variables
3. Polls the `SIU_RSR[XOSC]` bit until it indicates that the oscillator has locked
4. Switches the clock source to the crystal oscillator by setting `SIU_SYSDIV[SYSCLKSEL] = 0x1`
5. Disables the core watchdog timer and sets the SWT timeout period to  $2.5 \times 2^{27}$  cycles
6. Waits for the host to send a 64-bit password by either the CAN or SCI
7. Receives start address, size of download code in bytes, and VLE bit
8. Receives the application code data
9. Disables the SWT and enables the core watchdog timer with a timeout period of  $2.5 \times 2^{28}$  CPU clock cycles
10. Branches to the loaded code at the start address

The MMU setup depends on how the BAM entered serial boot mode. If it attempted to boot from the development bus but switched to serial boot because it could not find a valid RCHW in Flash, the MMU is set up as for that mode (see [Table 44-13](#)). Otherwise, the MMU setup matches [Table 44-2](#).

Serial boot mode can run in two modes, depending on the state of the SIU\_RSR[ABR] bit:

- Standard serial boot mode—fixed baud rates derived from the MCU system frequency. EVTO pin is kept high during reset.
- Baud Rate Detection serial boot mode—allows communication with adaptable speed, based on measured baud rate of the host transmission. EVTO pin has to be driven low for this mode.

In enhanced mode, the SIU\_RSR[ABR] bit is loaded from the ABR bit of the Miscellaneous DCF record. In legacy mode, the latching of the EVTO pin status in the SIU\_RSR register is done following ways:

- If no debug tool is attached, then the state of EVTO pin is latched upon negation of system reset
- If a debug tool, using EVTO pin, is attached, the state of EVTO pin is latched upon enabling the Nexus module

The EVTO pin is pulled up by an internal pull-up. Actively driven low EVTO pin selects the Baud Rate Detection mode.

When Fixed Baud Rate mode is selected, the BAM program configures the RXD\_A pin to be the input of the eSCI\_A module, CNRX\_A pin as an input and CNTX\_A as an output of the CAN\_A module. When Baud Rate Detection Mode is selected, the BAM program configures RXD\_A and CNRX\_A pins as GPI inputs for polling their state by the CPU.

The SCI and CAN controller pin configuration summaries are shown in [Table 44-8](#).

**Table 44-8. CAN/eSCI pin configuration for CAN/eSCI fixed baud rate boot modes**

| Pins   | Reset Function | Initial Serial Boot Mode |                                         | Serial Boot Mode after a valid CAN message received |                                         | Serial Boot Mode after a valid eSCI message received |                                         |
|--------|----------------|--------------------------|-----------------------------------------|-----------------------------------------------------|-----------------------------------------|------------------------------------------------------|-----------------------------------------|
|        |                | Function                 | Pad Configuration                       | Function                                            | Pad Configuration                       | Function                                             | Pad Configuration                       |
| CNTX_A | GPIO           | CNTX_A                   | Push/Pull output, with medium slew rate | CNTX_A                                              | Push/Pull output, with medium slew rate | GPIO                                                 | —                                       |
| CNRX_A | GPIO           | CNRX_A                   | Input with pullup and hysteresis        | CNRX_A                                              | Input with pullup and hysteresis        | GPIO                                                 | —                                       |
| TXD_A  | GPIO           | GPIO                     | —                                       | GPIO                                                | —                                       | TXD_A                                                | Push/Pull output, with medium slew rate |
| RXD_A  | GPIO           | RXD_A                    | Input with pullup and hysteresis        | GPIO                                                | —                                       | RXD_A                                                | Input with pullup and hysteresis        |

The BAM configures the communication modules for reception with fixed baud rates as shown in [Table 44-9](#) and then waits for data reception. The watchdog timers are configured as shown in [Table 44-10](#).

**Table 44-9. Serial boot fixed baud rates**

| Crystal frequency (MHz) | Peripheral clock frequency (MHz) | Desired SCI baud rate (baud) | Actual SCI baud rate (baud)     | CAN baud rate <sup>1</sup> (baud) |
|-------------------------|----------------------------------|------------------------------|---------------------------------|-----------------------------------|
| $f_{xtal}$              | $f_{per} = 0.5 \times f_{xtal}$  | —                            | $f_{per} \div 416$ <sup>2</sup> | $f_{per} \div 40$ <sup>3</sup>    |
| 8                       | 4                                | 9600                         | 9615                            | 100K                              |
| 12                      | 6                                | 14400                        | 14423                           | 150K                              |
| 16                      | 8                                | 19200                        | 19230                           | 200K                              |
| 20                      | 10                               | 24000                        | 24038                           | 250K                              |
| 40                      | 20                               | 48000                        | 48076                           | 500K                              |

1. The CAN engine is clocked by the system clock.
2. The transmitter clock frequency, which is denoted by the bit time, is determined by the equation  $f_{CLK} = f_{sys} \div (16 \times SBR) = SCI_{BaudRate}$  where SBR is a constant defined on BAM code to provide the correct baud rate.
3. The Baud Rate Division Factor is defined as  $BRF = (PRES DIV + 1) \times \text{Number of Time Quanta}$ , where the number of time quanta is dependent on the SYN\_REG, Time Segment 1, and Time Segment 2 constants. See [Figure 1](#).

**Table 44-10. Watchdog timeout settings for serial boot**

| Core WD                   |                                                                                    | SWT           |                                                                                    |
|---------------------------|------------------------------------------------------------------------------------|---------------|------------------------------------------------------------------------------------|
| Clock source              | Condition                                                                          | Clock source  | Condition                                                                          |
| XOSC $\div$ 2<br>4–20 MHz | Disabled during boot<br>Enabled after boot<br>Timeout = $2.5 \times 2^{28}$ cycles | IRC<br>16 MHz | Enabled during boot<br>Disabled after boot<br>Timeout = $2.5 \times 2^{27}$ cycles |

If a message containing 8 bytes with ID 0x11 is received by the CAN controller first, the BAM program:

- Transitions to serial CAN boot mode
- Disables the eSCI
- Configures RXD\_A to its reset state
- Transitions to the CAN serial download protocol routine

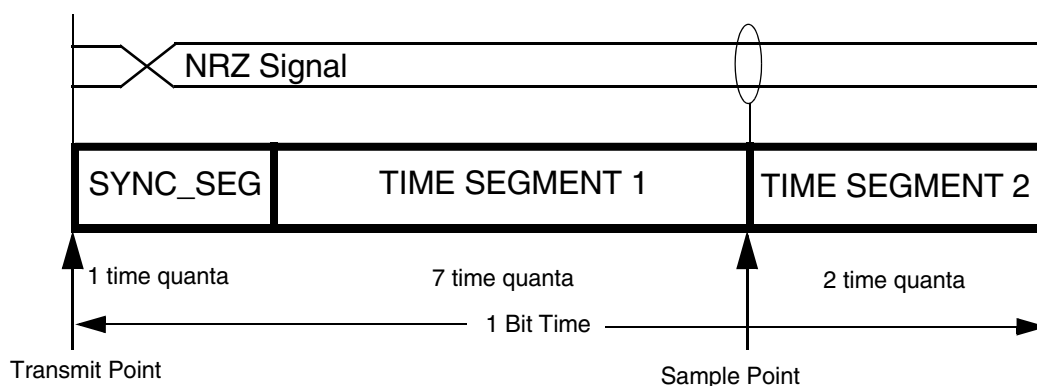
If a byte from eSCI is received first, the BAM program:

- Transitions to the Serial SCI Boot sub-mode
- Disables CAN\_A module
- Configures CAN\_A's pins to their reset state
- Transitions to the SCI serial download protocol routine

### 44.5.5.1 CAN controller configuration in fixed baud rate mode

The CAN controller is configured to operate at a baud rate equal to system frequency divided by 40, using the standard 11-bit identifier format detailed in the CAN 2.0A specification. See [Table 44-9](#) for examples of baud rates. Only one message buffer 0 is used for all communications.

The bit timing is configured as shown in the following figure.



Note: 1 Time quanta = 4 System clock periods

**Figure 44-2. CAN bit timing**

The BAM program ignores CAN errors and all received messages are assumed to be good. Received messages are transmitted back. The host processor must compare the MCU transmissions with the sent data and restart the process if an error is detected.

### 44.5.5.2 SCI controller configuration in fixed baud rate mode

The eSCI is configured for 1 start bit, 8 data bits, no parity, 1 stop bit and operates at a baud rate equal to system clock divided by 416. See [Table 44-9](#) for examples of baud rates.

The BAM program ignores the eSCI errors. All data received is assumed to be good and is sent back through the TXD pin. The host processor must compare the MCU transmitted data with the sent data and restart the process if an error is detected.

### 44.5.5.3 Serial boot mode download protocol

The download protocol follows four steps:

1. Host sends 64-bit password.
2. Host sends start address, size of download code in bytes, and VLE bit.
3. Host sends the application code data.

4. The MCU switches to the loaded code at the start address.

The communication is done in half-duplex manner; any transmission from host is followed by the MCU transmission. The host computer should not send data until it receives echo from the MCU. All multi-byte data structures must be sent most-significant byte (MSB) first.

When the CAN is used for serial download, the data is packed into standard CAN messages in the following manner:

- A message with 0x11 ID and 8-byte length to send the password. The MCU transmits back the same data, but with ID of 0x1.
- A message with 0x12 ID and 8-byte length to send the start address, length, and the VLE mode bit. The MCU transmits back the same data, but with ID of 0x2.
- Messages with 0x13 ID are used to send the downloaded data. The MCU transmits back the same data with ID of 0x3.

When the SCI is used for serial download, the data must be sent on a byte-by-byte basis. The MCU transmits back the received bytes.

### NOTE

Since the MCU starts up with the system clock derived from the Internal RC Oscillator (IRC), it does not wait for the crystal oscillator to stabilize before finishing the reset sequence and negating the  $\overline{\text{RTSTOUT}}$  pin. The BAM program starts executing with the IRC clock, but it needs to switch to the crystal oscillator (XOSC) clock in case of serial boot. Before switching to the XOSC clock, the BAM program waits for the SIU\_RSR[XOSC] bit to be asserted, which is an indication that the crystal oscillator has settled. As a consequence, the external device that will upload the serial code needs to wait enough time after  $\overline{\text{RSTOUT}}$  negates to allow the crystal oscillator to stabilize. The exact time depends on the crystal used, but is typically less than 10 ms. On the other hand, the code must be uploaded before the SWT timeout expires. The SWT timeout is 21 seconds.

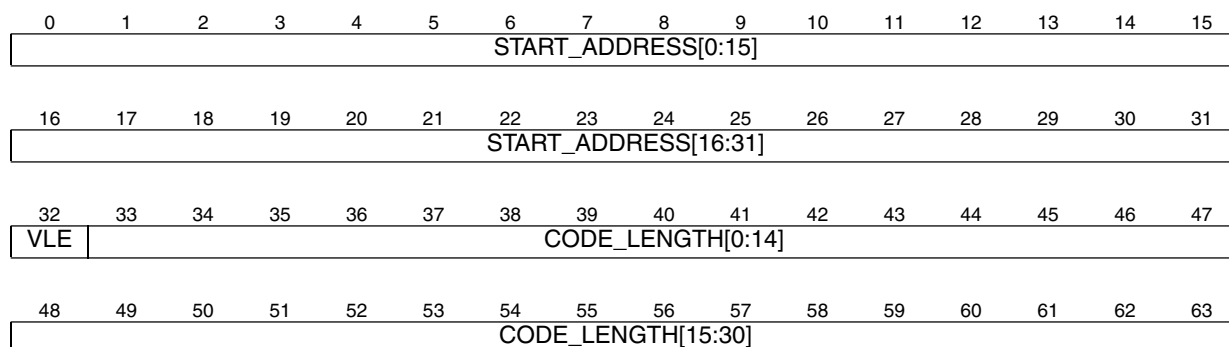
#### 44.5.5.4 Download protocol execution

The BAM program executes the serial boot according to the following steps:

1. Download 64-bit password.
  - The received 8-byte password is checked for validity. For a password to be valid, none of its four 16-bit half words must equal 0x0000 or 0xFFFF.
  - The password is compared to the fixed value of 0xFEED\_FACE\_CAFE\_BEEF.

- If the password check fails, the MCU stops responding. Assert RESET to force the MCU out of this state.
  - If the password check passes, the BAM transitions to the next step in the protocol.
2. Download start address, size of download, and VLE bit.

The BAM considers the next 8 bytes to contain a 32-bit start address, the VLE mode bit, and a 31-bit code length, as shown below.



- START\_ADDRESS defines where the received data is stored and where the MCU branches after the download is finished. The two least significant bits of the start address are ignored by the BAM program, thus the loaded code should be 32-bit word aligned.
  - CODE\_LENGTH defines how many data bytes to be loaded.
  - VLE instructs the MCU to program MMU entries 1–3 with VLE attribute. If it is 1, the downloaded code must be compiled to VLE instructions, if it is 0 the code contains classic Power Book E architecture instructions.
3. Download data.

Each byte of data received is stored in the MCU memory, starting at the START\_ADDRESS specified in the previous step. The data increments through memory until the number of bytes stored matches CODE\_LENGTH specified in the previous step.

The BAM program buffers incoming data, collecting up to 8 bytes. The buffered data is written to the RAM with 64-bit writes to prevent ECC errors, which may happen if the MCU RAM is protected by 64-bit ECC code.

Once the buffered data is written to the RAM, the BAM program refreshes the SWT.

### NOTE

Only system RAM supports 64-bit writes; therefore, attempting to download data to other RAM apart from system RAM causes errors.

If the start address of the downloaded data is not on an 8-byte boundary, the BAM writes 0x0 to the memory locations from the proceeding 8-byte boundary to the start address (maximum 4 bytes). The BAM also writes 0x0 to all memory locations from the last byte of data downloaded to the following 8 byte boundary (maximum 7 bytes). An additional 8 zero bytes are written to prevent possible ECC errors that may be caused by the CPU pre-fetching.

#### 4. Switch to the loaded code.

The BAM program waits for the last echo message transmission to complete, then the active communication controller is disabled. Its pins revert to GPIO inputs.

To provide compatibility with older devices, the BAM:

- Writes the core's time base registers (TBU and TBL) with 0x0
- Enables the core watchdog to cause a reset after a timeout period of  $2.5 \times 2^{28}$  CPU clock cycles
- Disables software watchdog (SWT)

The BAM code passes control to the loaded code at START\_ADDRESS, which was received in step 2 of the protocol.

### NOTE

The loaded code must periodically refresh the core watchdog timer or change the timeout period to a value that does not cause resets during normal operation.

#### 44.5.5.5 Baud rate detection procedure

When SIU\_RSR[ABR] is set, the BAM program enters into Baud Rate Detection mode.

The following additional steps are taken:

1. The baud rate detection routine is copied to the beginning of the system RAM from the BAM ROM to improve baud rate detection accuracy.
2. The CPU branches to the RAM.
3. The MCU configures the CNRX\_A and RXD\_A pins as general purpose inputs.
4. The MCU polls these GPIs until one of them goes low.

- If the CNRX\_A pin transitions first, the BAM program starts CAN baud rate detection routine, ignoring RXD\_A. After detecting the CAN baud rate, the BAM program transitions to the CAN download protocol routine described above.
- If the RXD\_A pin transitions first, the SCI baud rate detection and download protocol routines are called, ignoring any further CAN pins activity.

The host, executing the serial boot, should send a test frame so that the MCU can detect the host communication speed. When booting through the SCI, the host should send a zero byte as the test frame. When booting through the CAN, the host should send zero ID, zero length message as the test frame.

#### **44.5.5.5.1 SCI baud rate detection**

The host must send a zero byte to allow the MCU to detect the serial link baud rate. The host transmits one start bit, eight zero data bits, and one stop bit (the MCU does not echo it).

The MCU polls the RXD\_A pin for high-to-low transition and starts the e200 core Time Base counter (TB). Then the MCU polls for low-to-high transition on the pin and when it happens, the MCU turns off the TB counter. The TB content is used to calculate incoming signal baud-rate. The SCI baud rate is equal to the TB content divided by 288. (Measured over 9 bits with 16 system clocks per bit and the core frequency is two times higher than system frequency).

#### **44.5.5.6 CAN baud rate detection**

The host transmits a zero length message with zero 11 bit ID and MCU measures time over 40 bits, polling CNRX\_A pin for high and low, according to the sent data. The MCU does not acknowledge this message.

The CAN baud rate depends on the number of quantas per bit and serial clock frequency, which is defined by a prescaler. The CAN baud rate detection routine selects these parameters to maximize number of quantas per bit and achieve minimum difference between measured value and duration of the 40 CAN bits, to be programmed with selected pair of the parameters.



The CAN controller can be programmed with 8 to 25 number of quantas per bit. The bit timing parameters, which can be selected by the baud rate detection routine, are shown in the table below.

**Table 44-11. Lookup table for CAN bit timing**

| Time quanta per bit | Time segment 1 |       | Time Segment 2 | RJW |
|---------------------|----------------|-------|----------------|-----|
|                     | PROPSEG        | PSEG1 | PSEG2          |     |
| 8                   | 1              | 3     | 3              | 2   |
| 9                   | 2              | 3     | 3              | 2   |
| 10                  | 3              | 3     | 3              | 2   |
| 11                  | 4              | 3     | 3              | 2   |
| 12                  | 3              | 4     | 4              | 2   |
| 13                  | 4              | 4     | 4              | 3   |
| 14                  | 5              | 4     | 4              | 3   |
| 15                  | 6              | 4     | 4              | 4   |
| 16                  | 7              | 4     | 4              | 3   |
| 17                  | 8              | 4     | 4              | 3   |
| 18                  | 7              | 5     | 5              | 4   |
| 19                  | 8              | 5     | 5              | 4   |
| 20                  | 7              | 6     | 6              | 4   |
| 21                  | 8              | 6     | 6              | 4   |
| 22                  | 7              | 6     | 6              | 4   |
| 23                  | 8              | 6     | 6              | 4   |
| 24                  | 7              | 7     | 7              | 4   |
| 25                  | 8              | 7     | 7              | 4   |

Maximum and minimum speeds of the serial communication modules are defined by the MCU system frequency and shown in the table below.

**Table 44-12. Maximum and minimum detectable baud rates (crystal oscillator as reference clock source)**

| $F_{xtal}$ (MHz) | $F_{sys} = 0.5 \times F_{xtal}$ (MHz) | Max baud rate for CAN <sup>1</sup> (bps) | Min CAN baud rate ( $F_{sys}/25/256$ ) (bps) | Max baud rate for SCI ( $F_{sys}/160$ ) (bps) | Min baud rate for SCI ( $F_{sys}/160/2^{13}$ ) (bps) |
|------------------|---------------------------------------|------------------------------------------|----------------------------------------------|-----------------------------------------------|------------------------------------------------------|
| 8                | 4                                     | 1M                                       | 625                                          | 25K                                           | 3.0                                                  |
| 12               | 6                                     |                                          | 937.5                                        | 37.5K                                         | 4.6                                                  |
| 16               | 8                                     |                                          | 1250                                         | 50K                                           | 6.1                                                  |
| 20               | 10                                    |                                          | 1562.5                                       | 62.5K                                         | 7.6                                                  |
| 40               | 20                                    |                                          | 3125                                         | 125K                                          | 15.2                                                 |

1. Limited by 1 Mbps by CAN standard.

## 44.5.6 Booting from the development bus

If the MCU boots in one of the Development Bus boot modes, the BAM program:

- Reprograms the MMU entries for EBI and internal flash (see [Table 44-13](#))
- Sets up the EBI and development bus pins
- Tries to read RCHW from logical address 0x2000\_0000

If the valid RCHW is read from that address, the BAM program:

- Reads the user application code start address from 0x2000\_0004 address
- Parses RCHW
- Sets up watchdogs
- Updates EBI, SRAM, and internal flash MMU entries (1–3), according to RCHW[VLE]
- Passes control to the user code

If no valid RCHW was read, BAM switches to serial boot mode.

**Table 44-13. MMU configuration for development bus boot modes**

| TBL entry | Region         | Logical base address | Physical base address | Size  | Attributes                              |
|-----------|----------------|----------------------|-----------------------|-------|-----------------------------------------|
| 1         | Internal flash | 0x0000_0000          | 0x2000_0000           | 16 MB | Not guarded<br>Big endian<br>Global PID |
| 2         | EBI            | 0x2000_0000          | 0x2000_0000           | 16 MB | Not guarded<br>Big endian<br>Global PID |

### 44.5.6.1 EBI configuration for separate address and data development bus boot mode

The BAM program sets up EBI related registers as shown in the table below.

**Table 44-14. Development bus/EBI register settings for separate address data lines mode**

| Register    | Function | Value       | Comments                                                             |
|-------------|----------|-------------|----------------------------------------------------------------------|
| EBI_CAL_BR0 | —        | 0x2000_0803 | Sets the base address to 0x2000_0000, 16-bit wide bus, burst Inhibit |
| EBI_CAL_OR0 | —        | 0xFF80_00F0 | Set 15 wait states, 8MB                                              |

*Table continues on the next page...*

**Table 44-14. Development bus/EBI register settings for separate address data lines mode (continued)**

| Register              | Function                         | Value  | Comments                                           |
|-----------------------|----------------------------------|--------|----------------------------------------------------|
| SIU_PCR259–SIU_PCR293 | D_ADD[12:30],<br>D_ADD_DAT[0:15] | 0x0440 | Set pads to 20pf drive strength                    |
| SIU_PCR305–SIU_PCR307 | D_ADD[9:11]                      |        |                                                    |
| SIU_PCR256            | D_CS0                            | 0x443  | Set pads to 20pf drive strength and pull-up enable |
| SIU_PCR295            | D_WE0                            |        |                                                    |
| SIU_PCR297            | D_OE                             |        |                                                    |
| SIU_PCR298            | D_TS                             |        |                                                    |

RCHW[PS0] must be programmed to 1, since the development bus does not support 32-bit port size in that sub-mode.

#### 44.5.6.2 EBI configuration for multiplexed address and data development bus boot mode

The BAM program sets up EBI related registers as shown in the table below.

**Table 44-15. Development bus/EBI register settings multiplexed mode**

| Register                                                  | Function         | Value       | Comments                                                                                  |
|-----------------------------------------------------------|------------------|-------------|-------------------------------------------------------------------------------------------|
| The following registers are programmed when RCHW[PS0] = 1 |                  |             |                                                                                           |
| EBI_MCR                                                   | —                | 0x0000_0806 | AD multiplexed mode, small port size use D16–31                                           |
| EBI_CAL_BR0                                               | —                | 0x2000_0883 | Sets the base address to 0x2000_0000, 16-bit wide bus, AD multiplexed mode, burst Inhibit |
| EBI_CAL_OR0                                               | —                | 0xFF80_00F0 | Set 15 wait states, 8 MB                                                                  |
| SIU_PCR259–SIU_PCR262                                     | D_ADD[12:15],    | 0x440       | Set pads to 20 pf drive strength                                                          |
| SIU_PCR305–SIU_PCR307                                     | D_ADD[9:11]      |             |                                                                                           |
| SIU_PCR263–SIU_PCR277                                     | D_ADD_DAT[16:30] | 0x840       | Set pads to 20 pf drive strength                                                          |
| SIU_PCR257                                                | D_ADD_DAT[31]    |             |                                                                                           |
| SIU_PCR278–SIU_PCR293                                     | D_ADD_DAT[0:15]  | 0x440       | Set pads to 20 pf drive strength                                                          |
| SIU_PCR256                                                | D_CS0            | 0x443       | Set pads to 20 pf drive strength and pullup enable                                        |
| SIU_PCR295                                                | D_WE0            |             |                                                                                           |
| SIU_PCR297                                                | D_OE             |             |                                                                                           |
| SIU_PCR298                                                | D_TS             |             |                                                                                           |
| SIU_PCR299                                                | D_ALE            |             |                                                                                           |
| The following register is reprogrammed when RCHW[PS0] = 0 |                  |             |                                                                                           |
| EBI_CAL_BR0                                               | —                | 0x2000_0083 | Sets the base address to 0x2000_0000, 32-bit wide bus, AD multiplexed mode, burst Inhibit |

If the boot memory device is 32 bits wide, the RCHW must be programmed in two first half-words of the memory because when the MCU tries to read RCHW, the EBI is configured to use data lines 16–31 for 16-bit port accesses.

# Chapter 45

## System Status and Configuration Module (SSCM)

### 45.1 Chip-specific SSCM information

This chip has one instance of the SSCM. The SSCM is part of the reset and boot subsystem.

#### 45.1.1 SSCM and UTEST flash memory

The SSCM loads data stored in multiple locations of the UTEST portion of flash memory.

##### 45.1.1.1 Life Cycle (LC) slots

To determine the chip's Life Cycle status, the SSCM reads and interprets the LC slots according to the [Life Cycle](#) information. The SSCM\_LCSTAT[LC] field presents the results of the determination. The following table shows the LC slots' locations in UTEST flash memory.

**Table 45-1. Life Cycle slots in UTEST flash memory**

| Start address | End address | Bytes | Description                           |
|---------------|-------------|-------|---------------------------------------|
| 0040_0200h    | 0040_020Fh  | 16    | Life Cycle slot 0 - MCU Production    |
| 0040_0210h    | 0040_021Fh  | 16    | Life Cycle slot 1 - Customer Delivery |
| 0040_0220h    | 0040_022Fh  | 16    | Life Cycle slot 2 - OEM Production    |
| 0040_0230h    | 0040_023Fh  | 16    | Life Cycle slot 3 - In Field          |
| 0040_0240h    | 0040_024Fh  | 16    | Life Cycle slot 4 - Failure Analysis  |

### 45.1.1.2 System configuration data in DCF records and UTEST\_OFFSET location

Some DCF records in UTEST flash memory store data for system configuration. The SSCM scans these DCF records to provide information about the current state and configuration of the system.

On this chip, the UTEST\_OFFSET location at which the SSCM looks for DCF records is 0040\_0300h.

The following table summarizes these records. See the DCF Records chapter for additional details.

**Table 45-2. DCF records in UTEST flash memory scanned by SSCM**

| Address    | Purpose                                                                                                                     |
|------------|-----------------------------------------------------------------------------------------------------------------------------|
| 0040_0300h | STCU2 configuration<br>PASS configuration<br>Tamper Detect configuration<br>Miscellaneous configurations, such as boot mode |

### 45.1.1.3 Sources of errors flagged by SSCM\_STATUS[CER]

When set to 1, the **CER** field in the **SSCM System Status (SSCM\_STATUS)** register indicates that the SSCM has detected a configuration error during reset while loading initial system configuration.

On this chip, this field is set to 1 in these cases:

- When the process of loading any STCU2 DCF record encounters one of the conditions that generates a transfer error listed in the STCU2's [IPS bus interface](#) description
- When a DCF record that has the parity attribute is loaded without parity alignment

For a list of DCF clients and the corresponding DCF records on this chip, see [Table 5-6](#).

## 45.1.2 SSCM security reset and LC state

The SSCM provides a security mechanism to prevent device manipulation while loading the configuration data from dedicated slots in UTEST flash memory.

The SSCM ensures that the Life Cycle slots are correctly programmed and that at least one LC slot is valid. The Life Cycle state can only progress forward. Any attempt to move the LC state backward causes a security reset. This reset occurs in the cases where "System Reset" is the result of the scenarios in [Table 45-14](#).

### 45.1.3 SSCM\_ERROR[PAE] and abort responses to illegal accesses

The SSCM\_ERROR register's PAE field controls the abort responses to some illegal accesses to off-platform peripherals.

#### NOTE

This field has no effect on on-platform peripherals or on peripheral slots that are not allocated.

PAE, when set to 1, generates a transfer error in either of these cases of illegal accesses:

- The CPU tries to access any non-existing peripheral.
- The global module enable memory space is accessed.

Regardless of the PAE field, these cases always generate a transfer error:

- An access is attempted to a halted off-platform module.
- An accessed off-platform peripheral reports a transfer error.

## 45.2 Introduction

### 45.2.1 Overview

The System Status and Configuration Module (SSCM) is pictured in the figure below.

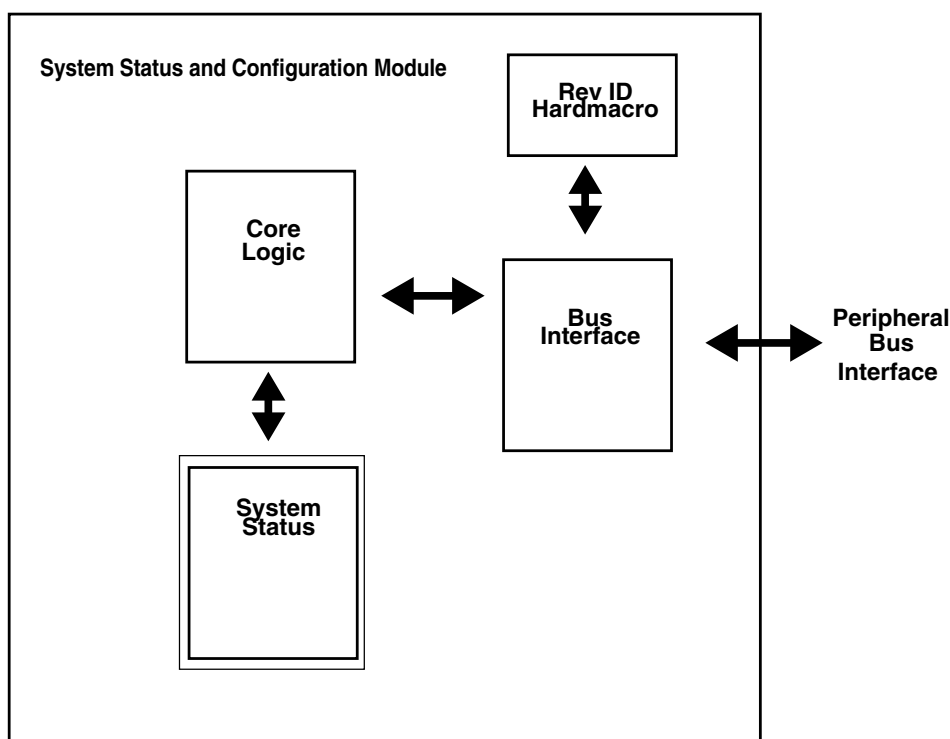


Figure 45-1. SSCM block diagram

## 45.2.2 Glossary

The table below shows a glossary of terms used through out the text.

Table 45-3. Glossary

| Term   | Definition                                                                                                                                                              |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0011b  | Binary numbers                                                                                                                                                          |
| 0Fh    | Hexadecimal numbers                                                                                                                                                     |
| Pin    | External physical connection.                                                                                                                                           |
| Signal | Electronic construct whose state or change in state conveys information.                                                                                                |
| X      | In certain contexts, such as a signal encoding, this indicates a don't care. For example, if a field is binary coded X001b, the state of the first bit is a don't care. |
| y      | y is used for a place holder of 1 hex digit for unknown values.                                                                                                         |

## 45.2.3 Features

The SSCM includes these distinct features:



- System Configuration and Status
  - Device Mode and System Status
  - Decodes the Device Life Cycle
- Peripheral abort enable/disable

### 45.2.4 Modes of operation

The SSCM operates identically in all system modes.

## 45.3 External signal description

The SSCM has no external pins.

## 45.4 Memory map and register definition

This section provides a detailed description of all memory-mapped registers in the SSCM.

The table below shows the memory map for the SSCM. Note that all addresses are offsets; the absolute address may be calculated by adding the specified offset to the base address of the SSCM.

#### NOTE

All registers are accessible via 8-bit, 16-bit or 32-bit reads and/or writes. However, 16-bit accesses must be aligned to 16-bit boundaries, and 32-bit accesses must be aligned to 32-bit boundaries. As an example, the STATUS register is accessible by a 16-bit read or write, SSCM Base + 02h, but performing a 16-bit access to SSCM Base + 0003h is illegal (that is, you should not make any transfer that is not aligned to burst size boundary).

#### NOTE

SSCM does not generate transfer errors in all memory holes.

**NOTE**

Reserved IPS register access is prohibited.

**SSCM memory map**

| Address offset (hex) | Register name                                  | Width (in bits) | Access | Reset value | Section/ page |
|----------------------|------------------------------------------------|-----------------|--------|-------------|---------------|
| 0                    | SSCM System Status (SSCM_STATUS)               | 16              | R      | See section | 45.4.1/2206   |
| 6                    | SSCM Error Configuration Register (SSCM_ERROR) | 16              | R/W    | 0000h       | 45.4.2/2208   |
| 34                   | Life Cycle Status Register (SSCM_LCSTAT)       | 32              | R      | See section | 45.4.3/2209   |

**45.4.1 SSCM System Status (SSCM\_STATUS)**

The System Status register reflects the current state of the system.

Address: 0h base + 0h offset = 0h

|       |          |     |     |       |      |    |          |    |
|-------|----------|-----|-----|-------|------|----|----------|----|
| Bit   | 0        | 1   | 2   | 3     | 4    | 5  | 6        | 7  |
| Read  | 0        | CER | MER | NXEN1 | NXEN | 1  | Reserved | 0  |
| Write |          | w1c | w1c |       |      |    |          |    |
| Reset | 0        | 0   | 0   | *     | *    | 1  | *        | 0  |
| Bit   | 8        | 9   | 10  | 11    | 12   | 13 | 14       | 15 |
| Read  | Reserved |     |     | 1     | 0    | 0  | 0        |    |
| Write |          |     |     |       |      |    |          |    |
| Reset | *        | *   | *   | 1     | 0    | 0  | 0        | 0  |

\* Notes:

- Reserved field: Each bit of this field can reset to 0 or 1.
- Reserved field: This field can reset to 0 or 1.
- NXEN field: Reset value depends on the device status after leaving reset.
- NXEN1 field: Reset value depends on the device status after leaving reset..

**SSCM\_STATUS field descriptions**

| Field         | Description                                                                                                                                                                                                                                                                                                                                                                                                            |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                |
| 1<br>CER      | Configuration Error<br><br>This field indicates that the SSCM has detected a configuration error during reset while loading initial device configuration. See the chip-specific SSCM information for details about sources of the error. If a non-permanent error is detected, this bit can be cleared by writing a 1.<br><br>0 No configuration problem detected by the SSCM<br>1 Device configuration is not correct |

Table continues on the next page...

**SSCM\_STATUS field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2<br>MER          | <p>Memory Configuration Error</p> <p>This field indicates that the SSCM has received notification of a memory configuration error. This error may have been detected during the reset sequence.</p> <p>If a detected error is not permanent, this field can be cleared by writing 1 to it.</p> <p>0 No memory configuration problem detected by the SSCM<br/>1 Device memory configuration is not correct</p> |
| 3<br>NXEN1        | <p>Processor 1 Nexus enable status</p> <p>0 Processor 1 Nexus disabled.<br/>1 Processor 1 Nexus enabled</p>                                                                                                                                                                                                                                                                                                   |
| 4<br>NXEN         | <p>Processor 0 Nexus enable status</p> <p>0 Processor 0 Nexus disabled.<br/>1 Processor 0 Nexus enabled</p>                                                                                                                                                                                                                                                                                                   |
| 5<br>Reserved     | <p>This field is reserved.<br/>This read-only field is reserved and always has the value 1.</p>                                                                                                                                                                                                                                                                                                               |
| 6<br>Reserved     | <p>This field is reserved.<br/>This field is for internal use. Users should ignore read values and should not attempt to write the field.</p>                                                                                                                                                                                                                                                                 |
| 7<br>Reserved     | <p>This field is reserved.<br/>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                               |
| 8–10<br>Reserved  | <p>This field is reserved.<br/>This field is for internal use. Users should ignore read values and should not attempt to write the field.</p>                                                                                                                                                                                                                                                                 |
| 11<br>Reserved    | <p>This field is reserved.<br/>This read-only field is reserved and always has the value 1.</p>                                                                                                                                                                                                                                                                                                               |
| 12<br>Reserved    | <p>This field is reserved.<br/>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                               |
| 13<br>Reserved    | <p>This field is reserved.<br/>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                               |
| 14–15<br>Reserved | <p>This field is reserved.<br/>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                               |

### 45.4.2 SSCM Error Configuration Register (SSCM\_ERROR)

The Error Configuration register is a read-write register that controls the error handling of the system.

Address: 0h base + 6h offset = 6h

|       |   |   |    |    |    |    |     |          |
|-------|---|---|----|----|----|----|-----|----------|
| Bit   | 0 | 1 | 2  | 3  | 4  | 5  | 6   | 7        |
| Read  | 0 |   |    |    |    |    |     |          |
| Write |   |   |    |    |    |    |     |          |
| Reset | 0 | 0 | 0  | 0  | 0  | 0  | 0   | 0        |
| Bit   | 8 | 9 | 10 | 11 | 12 | 13 | 14  | 15       |
| Read  | 0 |   |    |    |    |    | PAE | Reserved |
| Write |   |   |    |    |    |    |     |          |
| Reset | 0 | 0 | 0  | 0  | 0  | 0  | 0   | 0        |

SSCM\_ERROR field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–13<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 14<br>PAE        | Peripheral Bus Abort Enable.<br><br>This bit enables bus aborts on: <ul style="list-style-type: none"><li>any access to a peripheral slot that is not used on the device.<ul style="list-style-type: none"><li>This feature is intended to aid in debugging when developing application code.</li></ul></li><li>illegal accesses to off-platform peripherals.<ul style="list-style-type: none"><li>On platform peripherals cannot be controlled by SSCM peripheral abort.</li></ul></li></ul><br>0 Illegal accesses to non-existing peripherals do not produce a Prefetch or Data Abort exception<br>1 Illegal accesses to non-existing peripherals produce a Prefetch or Data Abort exception |
| 15<br>Reserved   | This field is reserved.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |

### 45.4.3 Life Cycle Status Register (SSCM\_LCSTAT)

Address: 0h base + 34h offset = 34h

|       |    |    |    |    |    |    |          |    |    |    |    |    |    |    |    |    |
|-------|----|----|----|----|----|----|----------|----|----|----|----|----|----|----|----|----|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6        | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0  |    |    |    |    |    |          |    |    |    |    |    |    |    |    |    |
| W     |    |    |    |    |    |    |          |    |    |    |    |    |    |    |    |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22       | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0  |    |    |    |    |    | Reserved |    | 0  |    |    |    |    |    | LC |    |
| W     |    |    |    |    |    |    |          |    |    |    |    |    |    |    |    |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | *        | *  | 0  | 0  | 0  | 0  | 0  | *  | *  | *  |

\* Notes:

- LC field: Reset value is the Life Cycle of the device, and is unaffected by reset.
- Reserved field: Reset Value is '11' for a valid secured part.

#### SSCM\_LCSTAT field descriptions

| Field             | Description                                                                                                                                                                                                                                     |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–21<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                         |
| 22–23<br>Reserved | This field is reserved.                                                                                                                                                                                                                         |
| 24–28<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                         |
| 29–31<br>LC       | Life Cycle<br>Determined by the SSCM by reading the LC slots. The encoding is:<br><br>000 Failure Analysis<br>001 Reserved<br>010 OEM Production<br>011 Customer Delivery<br>100 Reserved<br>101 Reserved<br>110 MCU Production<br>111 In Field |

## 45.5 Functional description

The primary purpose of the SSCM is to provide information about the current state and configuration of the system that may be useful for configuring application software and for debug of the system.

### 45.5.1 DCF mechanism

The DCF mechanism has been developed to handle settings of device parameters via OTP flash memory.

Starting at location UTEST\_OFFSET the user can store a series of DCF records - the device will process these records during the system reset sequence before the CPU leaves reset. The first record needs to be a start record followed by an application-specific number of data records as required, and finally a stop record.

Each record is 64 bits in length. [Table 45-8](#) shows the DCF start record. The start record must be placed at UTEST\_OFFSET to indicate to the device that the following data records should be processed.

**Table 45-8. DCF start record**

|            |
|------------|
| 0:31       |
| 05AA 55AFh |
| 0:31       |
| 0000 0000h |

The data records are structured similar to a CPU write instruction - 15 bits of client select, 15 bits of address, 32 bits of payload data plus a STOP bit - See [Table 45-9](#). In a data record the STOP bit must always be programmed to 0. Some clients support a parity bit.

**Table 45-9. Format of a DCF data record**

|             |            |      |      |
|-------------|------------|------|------|
| 0:31        |            |      |      |
| WDATA[31:0] |            |      |      |
| 0:14        | 15:29      | 30   | 31   |
| CS[14:0]    | ADDR[16:2] | PRTY | STOP |

The stop record indicates that processing should stop. The general format of the stop record is shown in [Table 45-10](#). Only the STOP bit needs to be 1 in order to form a stop record – All other bits are ignored. An unprogrammed location in UTEST flash will be interpreted as a stop record.

**Table 45-10. DCF stop record**

|      |    |
|------|----|
| 0:30 | 31 |
| rsvd | 1  |

*Table continues on the next page...*

**Table 45-10. DCF stop record (continued)**

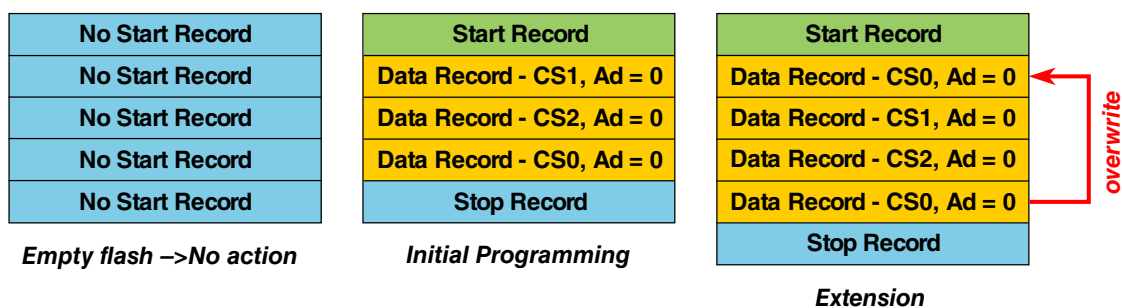
|      |
|------|
| 0:31 |
| rsvd |

If  $n$  data records are to be stored in UTEST, the data structure must be as shown in [Table 45-11](#).

**Table 45-11. Series of DCF records in UTEST**

| ADDR<br>offset | DATA        |            |      |        |
|----------------|-------------|------------|------|--------|
| 00h            | 05AA 55AFh  |            |      |        |
| 04h            | 0000 0000h  |            |      | STOP=0 |
| 08h            | WDATA[31:0] |            |      |        |
| 0Ch            | CS[14:0]    | ADDR[16:2] | PRTY | STOP=0 |
| 10h            | WDATA[31:0] |            |      |        |
| 14h            | CS[14:0]    | ADDR[16:2] | PRTY | STOP=0 |
| ...            | ...         |            |      |        |
| $8n - 1 + 0h$  | reserved    |            |      |        |
| $8n - 1 + 4h$  | reserved    |            |      | 1      |
| $8n + 0h$      |             |            |      |        |
| $8n + 4h$      |             |            |      |        |

There must never be an unprogrammed record in the data structure, as that would be interpreted as a stop record, so subsequent records would be ignored. This allows programming the records in several sessions, each time appending new records at the end of the list, as shown in [Figure 45-5](#).

**Figure 45-5. Appending DCF records**

In this case, a record may overwrite a client's value which was already set by a previous record. However whether such an operation is allowed depends on the rules given for the client - some clients are write-once, only allow setting bits but not clearing, and so on.

The parity bit (PRTY) is required for some DCF records. The parity scheme used is even parity, so the number of 1s in the WDATA and PRTY fields needs to be even. (For example, if the WDATA field has the value 0000\_0001h then the PRTY field needs to be set to 1, so that the total number of 1s is even.)

### NOTE

When programming DCF records, ensure that programming is not interrupted and can complete without error.

## 45.5.2 Life Cycle

The SSCM will determine the Life Cycle (LC) of the device by reading the LC slots from the UTEST flash memory. The read operation is done during the reset phase with normal timings and it is protected by operating monitors. In addition, a set of sanity checks executed over the LC read data guarantee the integrity of the final LC value.

At the end of the reset phase, the LC can have one of the following values:

- MCU Production
- Customer Delivery
- OEM Production
- In Field
- Failure Analysis

Life cycle can only be moved in the direction shown below:



**Figure 45-6. Life cycle sequence**

The LC is written into 5 slots, 128 bits each, at fixed locations in the UTEST flash memory block. Each LC slot is read in one single atomic operation and it is organized into two fields:

- the valid field
- the invalid field

Depending on the possible combination of the data programmed into these fields, each LC slot can have one of the possible 4 statuses as seen in [Table 45-12](#).



**Table 45-12. LC slot status**

| LC slots              |                         | LC slot value |
|-----------------------|-------------------------|---------------|
| Valid field (64 bits) | Invalid field (64 bits) |               |
| Erased                | Erased                  | Erased        |
| Marked                | Erased                  | Active        |
| Marked                | Marked                  | Inactive      |
| Any other values      |                         | Illegal       |

"Marked" in this case means that the value has been programmed with the bit pattern 55AA\_50AF\_55AA\_50AFh. "Erased" is detected by the bit pattern FFFF\_FFFF\_FFFF\_FFFFh. [Table 45-13](#) shows how the slots are arranged in memory.

**Table 45-13. LC slots in memory**

| Offset | LC slot word   |
|--------|----------------|
| 00h    | Valid[31:0]    |
| 04h    | Valid[63:32]   |
| 08h    | Invalid[31:0]  |
| 0Ch    | Invalid[63:32] |

The Life Cycle is determined as shown in [Table 45-14](#)

**NOTE**

"Erased" means the slot returns all 1s.

**Table 45-14. LC slots**

| LC slot 0<br>MCU Production | LC slot 1<br>Customer Delivery | LC slot 2<br>OEM Production | LC slot 3<br>In Field | LC slot 4<br>Failure Analysis | Resulting life cycle |
|-----------------------------|--------------------------------|-----------------------------|-----------------------|-------------------------------|----------------------|
| <b>Active</b>               | Erased                         | Erased                      | Erased                | Erased                        | MCU Production       |
| Inactive                    | <b>Active</b>                  | Erased                      | Erased                | Erased                        | Customer Delivery    |
| Inactive                    | Inactive                       | <b>Active</b>               | Erased                | Erased                        | OEM Production       |
| Inactive                    | Inactive                       | Inactive                    | <b>Active</b>         | Erased                        | In Field             |
| Inactive                    | Inactive                       | Inactive                    | Inactive              | <b>Active</b>                 | Failure Analysis     |
| Erased                      | Erased                         | Erased                      | Erased                | Erased                        | System Reset         |
| Illegal <sup>1</sup>        | Illegal                        | Illegal                     | Illegal               | Erased                        | In Field             |

1. Any value programmed other than shown in table above for Erased, Active, and Inactive would be treated as illegal.

**NOTE**

When programming Life Cycle, ensure that programming is not interrupted and can complete without error.

**NOTE**

Any other combination of values will result in a system reset.

**NOTE**

No one should program “ILLEGAL” even if able to do that.  
Device behavior is not guaranteed for these combinations.

## **45.6 Initialization and application information**

### **45.6.1 Reset**

The reset state of each individual bit appears in the detailed register descriptions.

## **45.7 Additional safety measures**

### **45.7.1 Spurious reset protection**

The SSCM implements protection against spurious module resets (for example, resets which only effect the SSCM, but not other modules) by gating the state machines status with the expected status of the system reset controller logic. If a spurious reset occurs, the SSCM will not interfere with the flash memory bus or overwrite configuration registers. However, internal SSCM status will be lost.

# Chapter 46

## Power Management

### 46.1 Overview

The MPC5775E microcontroller includes a robust power management infrastructure that enables applications to select among various operational modes and to monitor internal voltages for high- and low-voltage conditions. The monitoring capability is also used to ensure supply voltages and internal voltages are within the required ranges before the microcontroller can leave RESET.

The power management infrastructure consists of a Power Management Unit (PMU), which itself consists of a digital portion and an analog portion of a Power Management Controller (PMC).

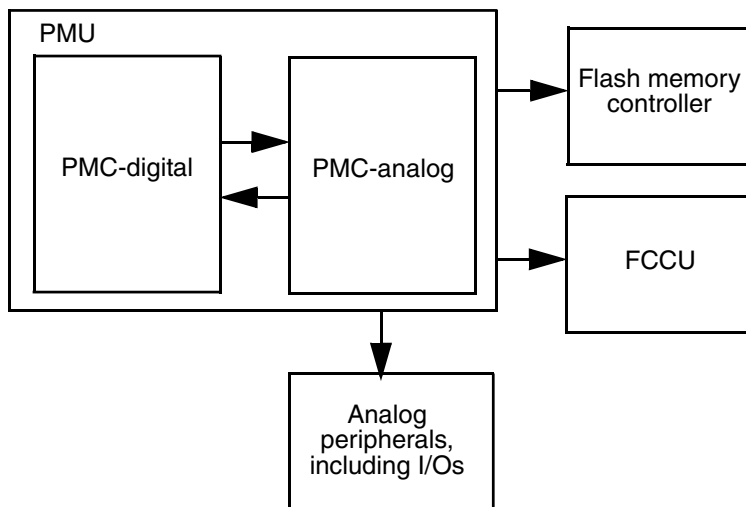


**Figure 46-1. Power management overview**

For details about any module, see the chapter dedicated to the module.

## 46.2 Functional description

The analog portion of the PMC provides a linear voltage regulator (LDO) and a switching mode regulator (SMPS) to generate the internal digital core supply and monitors various internal regulating points and supply voltages of the chip for functional safety purposes. The analog portion of the PMC includes an additional regulator for the flash memory block to provide the necessary 3.3 V source to the flash memory module.

The digital portion of the PMC contains the registers and digital logic to generate a power-on reset (POR) in response to particular voltage detection (VD) events and to execute a self test of the analog block. The PMC digital logic also contains the controls for PMC analog outputs to be sensed with the ADC module. In addition, the PMC digital logic contains the enable signals and other register controls for the Temperature Sensor modules.

The PMC connects to the Fault Collection and Control Unit (FCCU) and the flash memory controller. The on-chip ADCs receive signals from the PMC for diagnostic purposes.

## 46.3 Power modes

MPC5775E does not support ultra-low power modes that require power gating or separate power domains. The core logic implements stop and low frequency power saving modes, which may result in core current steps. To avoid undershoots or overshoots that trigger LVD or HVD events, ramp frequency changes smoothly.

## 46.4 External supply

The chip provides a bypass option that allows the core supply and flash memory supply to be supplied from an external source.

## 46.5 Power management supply description

See the product Data Sheet for supply descriptions.

## 46.6 Voltage monitoring

The PMC senses functional reset and supply voltages continuously to keep the chip in a safe operating range. When the voltage exceeds or falls below the limits defined by recommended electrical operating conditions, the PMC triggers a reset or, alternatively, an interrupt. See the PMC\_REE and PMC\_LVD\_HVD\_INTR\_ENB registers' descriptions for more information.



# Chapter 47

## Power Management Controller (PMC)

### 47.1 Introduction

The Power Management Controller (PMC) consists of two blocks: an analog power management block and a digital controller for the analog block.

The PMC digital controller contains the registers and digital logic to generate a POR in response to voltage-threshold events detected by the analog block. The PMC digital logic also contains:

- the controls for PMC analog outputs to be sensed with the ADC module.
- the enable signals and other registers for the Temperature Sensors.

#### NOTE

Temperature Sensor 0 and Temperature Sensor 1 each monitor two temperatures: temperature 0 refers to  $-40^{\circ}\text{C}$ , and temperature 2 refers to  $150^{\circ}\text{C}$ . Register field notation such as TEMP1\_0 designates Temperature Sensor 1's monitoring of temperature 0, or  $-40^{\circ}\text{C}$ .

#### 47.1.1 Features

The PMC module provides:

- 16 MHz clock frequency
- 32-bit IPS bus interface
- Memory-mapped configuration, control, and status registers
- Built-in self test for LVD and HVD during non-functional User mode

- Automatic loading of factory-defined trim/control information during POR for HVDs, LVDs, regulators, and voltage bandgap (via the DCF interface)
- Automatic loading of factory-defined trim configuration and control information to the analog PMC block
- Configuration and monitoring of the two Temperature Sensors
- Interrupt or POR generation for HVDs and LVDs (based on the configuration)
  - Option to generate interrupt, POR, or no action in response to:
    - LVD\_CORE\_COLD
    - HVD\_CORE
    - HVD\_HV
    - HVD\_FLASH
  - Automatic POR generation (except during PMC self test) in response to:
    - POR098\_c
    - LVD\_CORE\_HOT
    - POR\_HV
    - LVD\_HV
    - LVD\_FLASH
    - LVD\_IO
- Interrupt or POR generation for Temperature Sensors (based on the configuration)
- Fault injection to the FCCU for LVDs, HVDs, Temperature Sensors, and self test via IPS interface
- Precision LVD (low-voltage detection) and HVD (high-voltage detection)
- Bandgap voltage reference for:
  - LVDs and HVDs
  - ADC voltage reference channel
- FCCU: Software-triggered fault injection for LVDs, HVDs, Temperature Sensors, and LVD self test
- Self test logic for the LVDs and HVDs



## 47.2 Memory Map and Registers

Attempts to access 32-bit spaces starting at offset D4h result in a transfer error.

For *undocumented* 32-bit spaces at offsets lower than D4h:

- The reset value of the 32-bit space at offset B0h is 0000\_0080h.
- The reset value of every other 32-bit space is 0000\_0000h.
- Reads of and writes to these locations do not generate a transfer error, but the locations are reserved. Any writes to these locations must write the reset value.

**PMC memory map**

| Address offset (hex) | Register name                                                    | Width (in bits) | Access | Reset value | Section/ page                |
|----------------------|------------------------------------------------------------------|-----------------|--------|-------------|------------------------------|
| 88                   | Temperature Sensor Interrupt Enable Register (PMC_TS_IER)        | 32              | R/W    | 0000_0000h  | <a href="#">47.2.1/2222</a>  |
| 8C                   | Temperature Event Status Register (PMC_ESR_TD)                   | 32              | R/W    | 0000_0000h  | <a href="#">47.2.2/2224</a>  |
| 90                   | Temperature Reset Event Enable register (PMC_REE_TD)             | 32              | R/W    | 0000_003Fh  | <a href="#">47.2.3/2226</a>  |
| 94                   | Temperature Reset Event Selection Register (PMC_RES_TD)          | 32              | R/W    | 0000_0000h  | <a href="#">47.2.4/2228</a>  |
| 98                   | Temperature Detector Configuration Register (PMC_CTL_TD)         | 32              | R/W    | 0000_0303h  | <a href="#">47.2.5/2229</a>  |
| A4                   | Temperature Sensor Fault Injection Register (PMC_TS_FIR)         | 32              | W      | 0000_0000h  | <a href="#">47.2.6/2230</a>  |
| B4                   | Voltage Detect User Mode Test Register (PMC_SELF_TEST_UM_VD_REG) | 32              | R/W    | 0003_0000h  | <a href="#">47.2.7/2231</a>  |
| B8                   | FCCU Fault Injection Register (PMC_FIR)                          | 32              | W      | 0000_0000h  | <a href="#">47.2.8/2234</a>  |
| BC                   | LVD/HVD Interrupt Enable Control Register (PMC_LVD_HVD_INTR_ENB) | 32              | R/W    | 0000_0000h  | <a href="#">47.2.9/2235</a>  |
| C0                   | LVD/HVD Event Status Register (PMC_LVD_HVD_EVENT_STATUS)         | 32              | w1c    | 0000_0000h  | <a href="#">47.2.10/2238</a> |
| C4                   | ADC Channel Select Register (PMC_ADC_CS)                         | 32              | R/W    | 0000_0000h  | <a href="#">47.2.11/2240</a> |
| D0                   | Reset Event Enable Register (PMC_REE)                            | 32              | R/W    | 0000_00FFh  | <a href="#">47.2.12/2242</a> |

47.2.1 Temperature Sensor Interrupt Enable Register (PMC\_TS\_IER)

This configuration register contains a set of Interrupt Enable bits, each of which corresponds to a bit in the Temperature Event Status Register (ESR\_TD). These bits indicate whether an interrupt occurs when the temperature event occurs. A 0 indicates that no interrupt is to be sent, and a 1 indicates that an interrupt is to occur when the detected temperature event occurs. Enable these bits by writing 1 to the TS\_EN field of this register. After TS\_EN is 1, this register's Interrupt Enable bits can be read or written at any time.

Address: 0h base + 88h offset = 88h

|       |       |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|-------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0     | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | TS_EN | 0 |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |       |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0     | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |    |    |    |    |    |    |    |    |    |    |          |         |         |          |         |         |
|-------|----|----|----|----|----|----|----|----|----|----|----|----------|---------|---------|----------|---------|---------|
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27       | 28      | 29      | 30       | 31      |         |
| R     | 0  |    |    |    |    |    |    |    |    |    |    | Reserved | TS1_2IE | TS1_0IE | Reserved | TS0_2IE | TS0_0IE |
| W     |    |    |    |    |    |    |    |    |    |    |    |          |         |         |          |         |         |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0        | 0       | 0       | 0        | 0       |         |

PMC\_TS\_IER field descriptions

| Field            | Description                                                                                                                                                                                |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>TS_EN       | Temperature Sensor interrupts Enable<br><br>0    Temperature sensor interrupts are not enabled.<br>1    Enable temperature sensor interrupts.                                              |
| 1–25<br>Reserved | Reserved<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                    |
| 26<br>Reserved   | Reserved<br><br>This field is reserved.                                                                                                                                                    |
| 27<br>TS1_2IE    | Temperature Sensor 1 input 2 Interrupt Enable<br><br>This bit determines whether an interrupt is seen by the system when the temperature rises above its corresponding threshold (flag 2). |

Table continues on the next page...

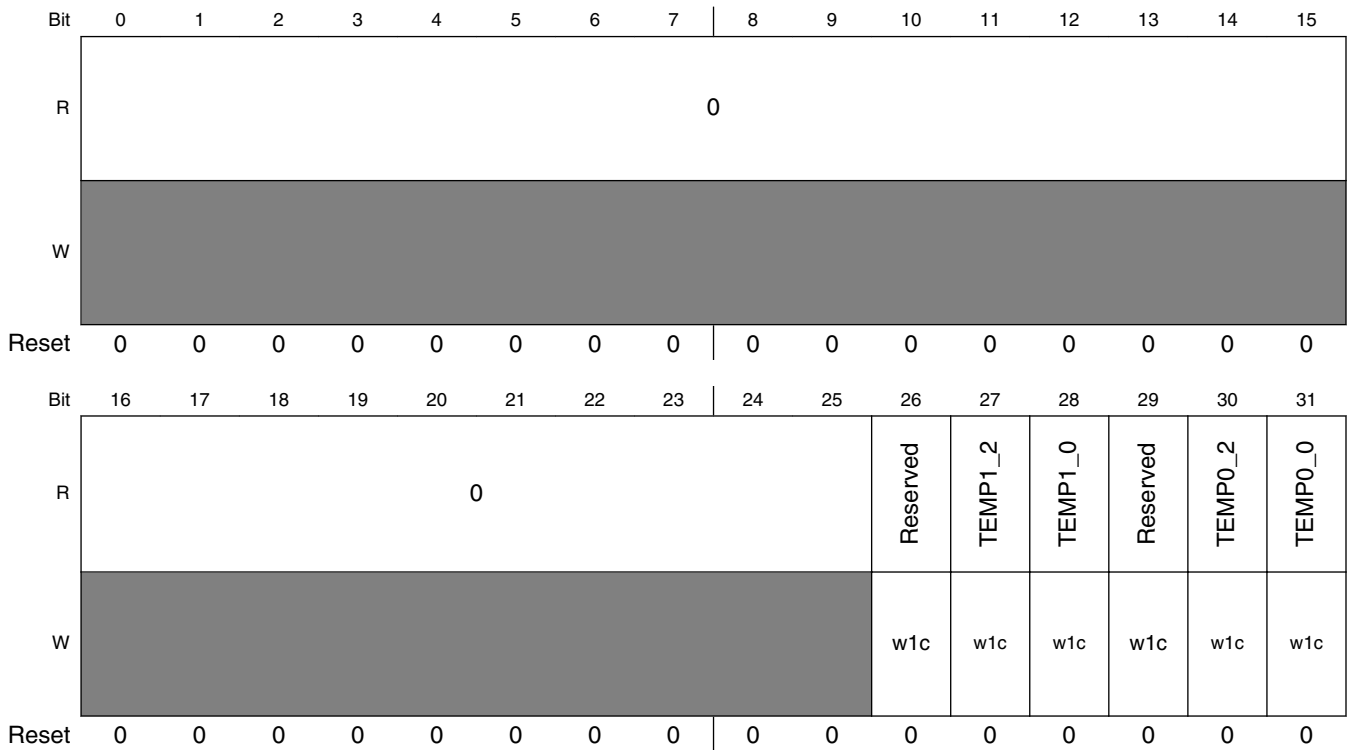
**PMC\_TS\_IER field descriptions (continued)**

| <b>Field</b>   | <b>Description</b>                                                                                                                                                                                                                       |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | 0 No interrupt.<br>1 Interrupt occurs.                                                                                                                                                                                                   |
| 28<br>TS1_0IE  | Temperature Sensor 1 input 0 Interrupt Enable<br><br>This bit determines whether an interrupt is seen by the system when the temperature falls below its corresponding threshold.<br><br>0 No interrupt.<br>1 Interrupt occurs.          |
| 29<br>Reserved | Reserved<br><br>This field is reserved.                                                                                                                                                                                                  |
| 30<br>TS0_2IE  | Temperature Sensor 0 input 2 Interrupt Enable<br><br>This bit determines whether an interrupt is seen by the system when the temperature rises above its corresponding threshold (flag 2).<br><br>0 No interrupt.<br>1 Interrupt occurs. |
| 31<br>TS0_0IE  | Temperature Sensor 0 input 0 Interrupt Enable<br><br>This bit determines whether an interrupt is seen by the system when the temperature falls below its corresponding threshold.<br><br>0 No interrupt.<br>1 Interrupt occurs.          |

47.2.2 Temperature Event Status Register (PMC\_ESR\_TD)

This register indicates the state of the two Temperature Sensor modules' signals. When a temperature event occurs, the corresponding flag bit sets to 1. To clear a flag bit to 0, write 1 to it.

Address: 0h base + 8Ch offset = 8Ch



PMC\_ESR\_TD field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                               |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–25<br>Reserved | Reserved<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                   |
| 26<br>Reserved   | Reserved<br><br>This field is reserved.                                                                                                                                                                                                                                                                                                                                                                                   |
| 27<br>TEMP1_2    | Temperature Sensor 1 event flag 2<br><br>This bit is associated with the temperature for the high temperature sensor 2 point (150°C). It sets to 1 when the temperature exceeds its corresponding threshold, and it clears to 0 when 1 is written to this bit location.<br><br>When this bit is 1: <ul style="list-style-type: none"><li>If the TS_IER[TS1_2IE] bit is also 1, an interrupt is sent to the CPU.</li></ul> |

Table continues on the next page...

## PMC\_ESR\_TD field descriptions (continued)

| Field          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | <ul style="list-style-type: none"> <li>If REE_TD[TEMP1_2] is also 1, a system reset is generated, which clears TS_IER[TS1_2IE] and negates the interrupt request.</li> <li>If REE_TD[TEMP1_2] is 1 <i>and</i> RES_TD[TEMP1_2] is 0, a POR is generated.</li> </ul> <p><b>NOTE:</b> If this flag is cleared to 0 while the over-temperature condition still exists, the flag will not automatically reassert to 1 until after the device falls below the corresponding over-temperature threshold.</p> <p>0 Currently no occurrence.<br/>1 Temperature occurrence detected.</p>                                                                                                                                                                                                                                                                                                                                                                                           |
| 28<br>TEMP1_0  | <p>Temperature Sensor 1 event flag 0</p> <p>This bit is associated with the temperature for the cold temperature sensor 0 point (–40°C). It sets to 1 when the temperature falls below its threshold, and it clears to 0 when 1 is written to this bit location.</p> <p>When this bit is 1:</p> <ul style="list-style-type: none"> <li>If the TS_IER[TS1_0IE] bit is also 1, an interrupt is sent to the CPU.</li> <li>If REE_TD[TEMP1_0] is also 1, a system reset is generated, which clears TS_IER[TS1_0IE] and negates the interrupt request.</li> <li>If REE_TD[TEMP1_0] is 1 <i>and</i> RES_TD[TEMP1_0] is 0, a POR is generated.</li> </ul> <p><b>NOTE:</b> If this flag is cleared to 0 while the under-temperature condition still exists, the flag will not automatically reassert to 1 until after the device rises above the under-temperature threshold.</p> <p>0 Currently no occurrence.<br/>1 Temperature occurrence detected.</p>                       |
| 29<br>Reserved | <p>Reserved</p> <p>This field is reserved.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 30<br>TEMP0_2  | <p>Temperature Sensor 0 event flag 2</p> <p>This bit is associated with the temperature for the high temperature sensor 2 point (150°C). It sets to 1 when the temperature exceeds its corresponding threshold, and it clears to 0 when 1 is written to this bit location.</p> <p>When this bit is 1:</p> <ul style="list-style-type: none"> <li>If the TS_IER[TS0_2IE] bit is also 1, an interrupt is sent to the CPU.</li> <li>If REE_TD[TEMP0_2] is also 1, a system reset is generated, which clears TS_IER[TS0_2IE] and negates the interrupt request.</li> <li>If REE_TD[TEMP0_2] is 1 <i>and</i> RES_TD[TEMP0_2] is 0, a POR is generated.</li> </ul> <p><b>NOTE:</b> If this flag is cleared to 0 while the over-temperature condition still exists, the flag will not automatically reassert to 1 until after the device falls below the corresponding over-temperature threshold.</p> <p>0 Currently no occurrence.<br/>1 Temperature occurrence detected.</p> |
| 31<br>TEMP0_0  | <p>Temperature Sensor 0 event flag 0</p> <p>This bit is associated with the temperature for the cold temperature sensor 0 point (–40°C). It sets to 1 when the temperature falls below its threshold, and it clears to 0 when 1 is written to this bit location.</p> <p>When this bit is 1:</p> <ul style="list-style-type: none"> <li>If the TS_IER[TS0_0IE] bit is also 1, an interrupt is sent to the CPU.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

*Table continues on the next page...*

## PMC\_ESR\_TD field descriptions (continued)

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | <ul style="list-style-type: none"> <li>If REE_TD[TEMPO_0] is also 1, a system reset is generated, which clears TS_IER[TS0_0IE] and negates the interrupt request.</li> <li>If REE_TD[TEMPO_0] is 1 <i>and</i> RES_TD[TEMPO_0] is 0, a POR is generated.</li> </ul> <p><b>NOTE:</b> If this flag is cleared to 0 while the under-temperature condition still exists, the flag will not automatically reassert to 1 until after the device rises above the under-temperature threshold.</p> <p>0 Currently no occurrence.<br/>1 Temperature occurrence detected.</p> |

## 47.2.3 Temperature Reset Event Enable register (PMC\_REE\_TD)

This register controls whether the temperature detect signal event causes a reset. If the desired flag bit is enabled, the temperature detect event can cause a reset to be generated when the selected temperature passes the trigger event.

**NOTE**

These bits can be loaded from a DCF record in the UTEST portion of the flash memory. When they are loaded from a DCF record, they cannot be changed by a register write.

Address: 0h base + 90h offset = 90h

|       |    |    |    |    |    |    |    |    |    |    |          |    |         |    |         |    |          |  |         |  |         |  |
|-------|----|----|----|----|----|----|----|----|----|----|----------|----|---------|----|---------|----|----------|--|---------|--|---------|--|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10       | 11 | 12      | 13 | 14      | 15 |          |  |         |  |         |  |
| R     | 0  |    |    |    |    |    |    |    |    |    |          |    |         |    |         |    |          |  |         |  |         |  |
| W     |    |    |    |    |    |    |    |    |    |    |          |    |         |    |         |    |          |  |         |  |         |  |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0        | 0  | 0       | 0  | 0       | 0  |          |  |         |  |         |  |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26       | 27 | 28      | 29 | 30      | 31 |          |  |         |  |         |  |
| R     | 0  |    |    |    |    |    |    |    |    |    | Reserved |    | TEMP1_2 |    | TEMP1_0 |    | Reserved |  | TEMP0_2 |  | TEMP0_0 |  |
| W     |    |    |    |    |    |    |    |    |    |    | Reserved |    | TEMP1_2 |    | TEMP1_0 |    | Reserved |  | TEMP0_2 |  | TEMP0_0 |  |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1        | 1  | 1       | 1  | 1       | 1  |          |  |         |  |         |  |

## PMC\_REE\_TD field descriptions

| Field            | Description                                                                                             |
|------------------|---------------------------------------------------------------------------------------------------------|
| 0–25<br>Reserved | Reserved<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 26<br>Reserved   | Reserved                                                                                                |

Table continues on the next page...

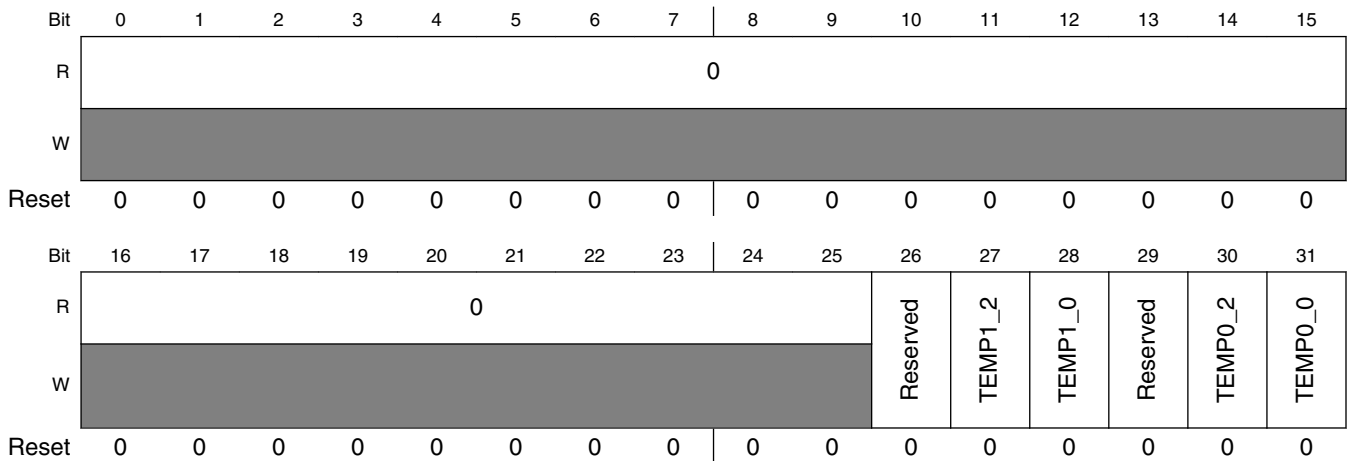
**PMC\_REE\_TD field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                                                                                                                                                   |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | This field is reserved.                                                                                                                                                                                                                                                                                                                                                                       |
| 27<br>TEMP1_2  | <p>Temperature sensor 1 detect assertion reset enable 2</p> <p>This bit defines whether a temperature detect assertion can generate a system reset. When this bit is 1, the RES_TD[TEMP1_2] bit determines whether the assertion causes a POR.</p> <p>0 Disabled. Temperature Sensor detect does not cause system reset.<br/>1 Enabled. Temperature Sensor detect can cause system reset.</p> |
| 28<br>TEMP1_0  | <p>Temperature sensor 1 detect assertion reset enable 0</p> <p>This bit defines whether a temperature detect assertion can generate a system reset. When this bit is 1, the RES_TD[TEMP1_0] bit determines whether the assertion causes a POR.</p> <p>0 Disabled. Temperature Sensor detect does not cause system reset.<br/>1 Enabled. Temperature Sensor detect can cause system reset.</p> |
| 29<br>Reserved | <p>Reserved</p> <p>This field is reserved.</p>                                                                                                                                                                                                                                                                                                                                                |
| 30<br>TEMP0_2  | <p>Temperature sensor 0 detect assertion reset enable 2</p> <p>This bit defines whether a temperature detect assertion can generate a system reset. When this bit is 1, the RES_TD[TEMP0_2] bit determines whether the assertion causes a POR.</p> <p>0 Disabled. Temperature Sensor detect does not cause system reset.<br/>1 Enabled. Temperature Sensor detect can cause system reset.</p> |
| 31<br>TEMP0_0  | <p>Temperature sensor 0 detect assertion reset enable 0</p> <p>This bit defines whether a temperature detect assertion can generate a system reset. When this bit is 1, the RES_TD[TEMP0_0] bit determines whether the assertion causes a POR.</p> <p>0 Disabled. Temperature Sensor detect does not cause system reset.<br/>1 Enabled. Temperature Sensor detect can cause system reset.</p> |

47.2.4 Temperature Reset Event Selection Register (PMC\_RES\_TD)

This register controls whether the temperature sensor detect signal event causes a POR or no reset. If the desired flag bit is enabled, the temperature sensor event causes either a POR or no reset to be generated when the selected temperature passes the trigger event.

Address: 0h base + 94h offset = 94h



PMC\_RES\_TD field descriptions

| Field            | Description                                                                                                                                                                     |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–25<br>Reserved | Reserved<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                         |
| 26<br>Reserved   | Reserved<br><br>This field is reserved.                                                                                                                                         |
| 27<br>TEMP1_2    | TEMP1_2 Reset Event Select<br><br>0   POR generated. Temperature Sensor assertion causes a POR.<br>1   No reset generated. Temperature Sensor assertion does not cause a reset. |
| 28<br>TEMP1_0    | TEMP1_0 Reset Event Select<br><br>0   POR generated. Temperature Sensor assertion causes a POR.<br>1   No reset generated. Temperature Sensor assertion does not cause a reset. |
| 29<br>Reserved   | Reserved<br><br>This field is reserved.                                                                                                                                         |
| 30<br>TEMP0_2    | TEMP0_2 Reset Event Select<br><br>0   POR generated. Temperature Sensor assertion causes a POR.<br>1   No reset generated. Temperature Sensor assertion does not cause a reset. |

Table continues on the next page...



**PMC\_RES\_TD field descriptions (continued)**

| Field         | Description                                                                                                                                                                 |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31<br>TEMP0_0 | TEMP0_0 Reset Event Select<br><br>0 POR generated. Temperature Sensor assertion causes a POR.<br>1 No reset generated. Temperature Sensor assertion does not cause a reset. |

**47.2.5 Temperature Detector Configuration Register (PMC\_CTL\_TD)**

Address: 0h base + 98h offset = 98h

|       |    |          |    |    |    |    |          |        |    |          |    |    |    |    |          |        |
|-------|----|----------|----|----|----|----|----------|--------|----|----------|----|----|----|----|----------|--------|
| Bit   | 0  | 1        | 2  | 3  | 4  | 5  | 6        | 7      | 8  | 9        | 10 | 11 | 12 | 13 | 14       | 15     |
| R     | 0  |          |    |    |    |    |          |        |    |          |    |    |    |    |          |        |
| W     |    |          |    |    |    |    |          |        |    |          |    |    |    |    |          |        |
| Reset | 0  | 0        | 0  | 0  | 0  | 0  | 0        | 0      | 0  | 0        | 0  | 0  | 0  | 0  | 0        | 0      |
| Bit   | 16 | 17       | 18 | 19 | 20 | 21 | 22       | 23     | 24 | 25       | 26 | 27 | 28 | 29 | 30       | 31     |
| R     | 0  | Reserved |    |    |    |    | Reserved | TS1_EN | 0  | Reserved |    |    |    |    | Reserved | TS0_EN |
| W     |    |          |    |    |    |    |          |        |    |          |    |    |    |    |          |        |
| Reset | 0  | 0        | 0  | 0  | 0  | 0  | 1        | 1      | 0  | 0        | 0  | 0  | 0  | 0  | 1        | 1      |

**PMC\_CTL\_TD field descriptions**

| Field             | Description                                                                                                                                                                                                                     |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–16<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                         |
| 17–21<br>Reserved | This field is reserved.                                                                                                                                                                                                         |
| 22<br>Reserved    | This field is reserved.                                                                                                                                                                                                         |
| 23<br>TS1_EN      | Temperature Sensor 1 Enable<br><br><b>NOTE:</b> If this bit is 0, all reset and interrupt events from Temperature Sensor 1 are disabled.<br><br>0 Disable Temperature Sensor 1 module.<br>1 Enable Temperature Sensor 1 module. |
| 24<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                         |
| 25–29<br>Reserved | This field is reserved.                                                                                                                                                                                                         |

*Table continues on the next page...*

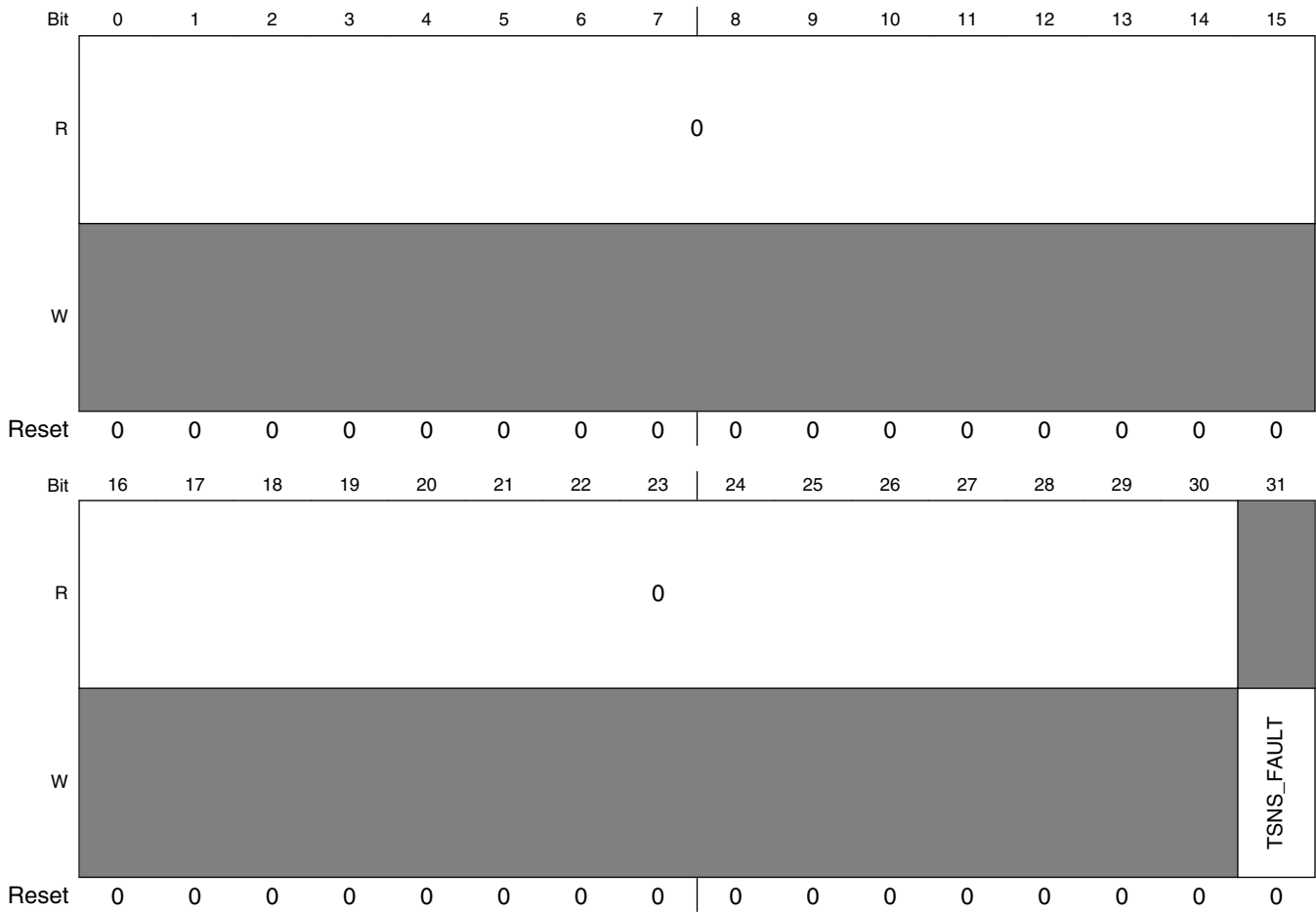
PMC\_CTL\_TD field descriptions (continued)

| Field          | Description                                                                                                                                                                                                                         |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 30<br>Reserved | This field is reserved.                                                                                                                                                                                                             |
| 31<br>TS0_EN   | Temperature Sensor 0 Enable<br><br><b>NOTE:</b> If this bit is 0, all reset and interrupt events from Temperature Sensor 0 are disabled.<br><br>0   Disable Temperature Sensor 0 module.<br>1   Enable Temperature Sensor 0 module. |

47.2.6 Temperature Sensor Fault Injection Register (PMC\_TS\_FIR)

This FCCU Fault Injection register enables user software to artificially generate FCCU faults corresponding to TEMP SENS. Writing 1 to one of these bits generates a single cycle pulsed fault to the FCCU block. The bit clears to 0 automatically in the next cycle.

Address: 0h base + A4h offset = A4h



**PMC\_TS\_FIR field descriptions**

| Field            | Description                                                                                                                                                                    |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–30<br>Reserved | Reserved<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                        |
| 31<br>TSNS_FAULT | Temperature Sensor Fault injection.<br><br>A write to this bit sends a FCCU fault (single cycle) for TEMP SENSOR.<br><br>0 No fault is sent.<br>1 Single cycle fault injected. |

### 47.2.7 Voltage Detect User Mode Test Register (PMC\_SELF\_TEST\_UM\_VD\_REG)

This register is used for testing LVDs and HVDs during user mode.

#### NOTE

A PMC self test of any or all LVDs and HVDs can be interrupted by these reset sources and thereby become corrupted:

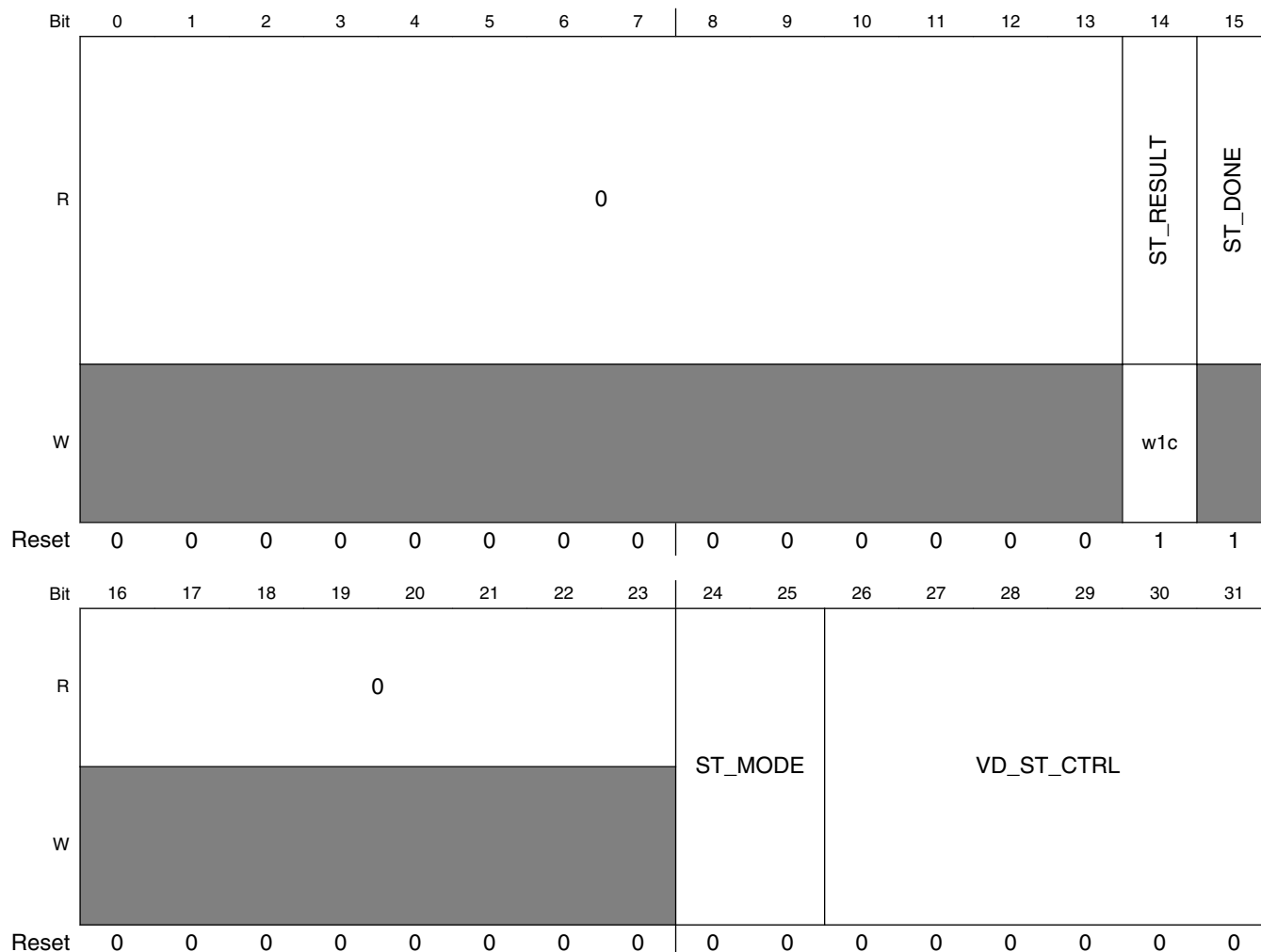
- Writing 1 to SIU\_SRCR[SSR]
- JTAG reset
- FCCU reset

Due to this corruption, the PMC or FCCU might report incorrectly that a PMC self test of any or all LVDs and HVDs failed.

If software detects one of these reset sources when exiting reset and the PMC or FCCU reports a failed PMC self test of any or all LVDs and HVDs, execute the PMC self test again to confirm the failure.

## Memory Map and Registers

Address: 0h base + B4h offset = B4h



### PMC\_SELF\_TEST\_UM\_VD\_REG field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                             |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–13<br>Reserved | Reserved<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                 |
| 14<br>ST_RESULT  | Self Test Result status<br>Valid only when ST_DONE is 1. It is cleared on reset or by the host writing 1.<br><br>0 Self Test Failed<br>1 Self Test Passed                                                                                                                                                                                               |
| 15<br>ST_DONE    | Self Test Done status<br>Self Test completed indication status. This bit is set to 1 (logic high) to indicate that a self test has completed. A value of 1 does not represent whether a self test has passed or failed. This bit is cleared to 0 automatically every time a self test is triggered.<br><br>0 Self test running<br>1 Self test completed |

Table continues on the next page...

**PMC\_SELF\_TEST\_UM\_VD\_REG field descriptions (continued)**

| Field               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                 |      |        |                 |        |          |        |          |        |          |        |          |        |               |        |              |        |          |        |        |        |          |        |        |        |        |        |           |        |           |        |        |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|------|--------|-----------------|--------|----------|--------|----------|--------|----------|--------|----------|--------|---------------|--------|--------------|--------|----------|--------|--------|--------|----------|--------|--------|--------|--------|--------|-----------|--------|-----------|--------|--------|
| 16–23<br>Reserved   | Reserved<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                 |      |        |                 |        |          |        |          |        |          |        |          |        |               |        |              |        |          |        |        |        |          |        |        |        |        |        |           |        |           |        |        |
| 24–25<br>ST_MODE    | Self Test Mode bits for testing LVDs and HVDs<br><br>00 Default<br>01 Software triggered self test.<br>10 Single VD test as indicated by the VD_ST_CTRL field.<br>11 Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                 |      |        |                 |        |          |        |          |        |          |        |          |        |               |        |              |        |          |        |        |        |          |        |        |        |        |        |           |        |           |        |        |
| 26–31<br>VD_ST_CTRL | Voltage Detect Self Test Control<br>User mode Test bits for testing of LVDs and HVDs. Encoding values not listed are reserved. <table border="1"> <thead> <tr> <th>VD_ST_CTRL[5:0]</th><th>Test</th></tr> </thead> <tbody> <tr><td>000000</td><td>Functional Mode</td></tr> <tr><td>000001</td><td>Reserved</td></tr> <tr><td>000010</td><td>POR098_c</td></tr> <tr><td>000011</td><td>Reserved</td></tr> <tr><td>000100</td><td>Reserved</td></tr> <tr><td>000101</td><td>LVD_CORE_COLD</td></tr> <tr><td>000110</td><td>LVD_CORE_HOT</td></tr> <tr><td>000111</td><td>HVD_CORE</td></tr> <tr><td>001000</td><td>POR_HV</td></tr> <tr><td>001001</td><td>Reserved</td></tr> <tr><td>001010</td><td>LVD_HV</td></tr> <tr><td>001011</td><td>HVD_HV</td></tr> <tr><td>001100</td><td>LVD_FLASH</td></tr> <tr><td>001101</td><td>HVD_FLASH</td></tr> <tr><td>001110</td><td>LVD_IO</td></tr> </tbody> </table> | VD_ST_CTRL[5:0] | Test | 000000 | Functional Mode | 000001 | Reserved | 000010 | POR098_c | 000011 | Reserved | 000100 | Reserved | 000101 | LVD_CORE_COLD | 000110 | LVD_CORE_HOT | 000111 | HVD_CORE | 001000 | POR_HV | 001001 | Reserved | 001010 | LVD_HV | 001011 | HVD_HV | 001100 | LVD_FLASH | 001101 | HVD_FLASH | 001110 | LVD_IO |
| VD_ST_CTRL[5:0]     | Test                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |                 |      |        |                 |        |          |        |          |        |          |        |          |        |               |        |              |        |          |        |        |        |          |        |        |        |        |        |           |        |           |        |        |
| 000000              | Functional Mode                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |                 |      |        |                 |        |          |        |          |        |          |        |          |        |               |        |              |        |          |        |        |        |          |        |        |        |        |        |           |        |           |        |        |
| 000001              | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                 |      |        |                 |        |          |        |          |        |          |        |          |        |               |        |              |        |          |        |        |        |          |        |        |        |        |        |           |        |           |        |        |
| 000010              | POR098_c                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                 |      |        |                 |        |          |        |          |        |          |        |          |        |               |        |              |        |          |        |        |        |          |        |        |        |        |        |           |        |           |        |        |
| 000011              | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                 |      |        |                 |        |          |        |          |        |          |        |          |        |               |        |              |        |          |        |        |        |          |        |        |        |        |        |           |        |           |        |        |
| 000100              | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                 |      |        |                 |        |          |        |          |        |          |        |          |        |               |        |              |        |          |        |        |        |          |        |        |        |        |        |           |        |           |        |        |
| 000101              | LVD_CORE_COLD                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                 |      |        |                 |        |          |        |          |        |          |        |          |        |               |        |              |        |          |        |        |        |          |        |        |        |        |        |           |        |           |        |        |
| 000110              | LVD_CORE_HOT                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                 |      |        |                 |        |          |        |          |        |          |        |          |        |               |        |              |        |          |        |        |        |          |        |        |        |        |        |           |        |           |        |        |
| 000111              | HVD_CORE                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                 |      |        |                 |        |          |        |          |        |          |        |          |        |               |        |              |        |          |        |        |        |          |        |        |        |        |        |           |        |           |        |        |
| 001000              | POR_HV                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                 |      |        |                 |        |          |        |          |        |          |        |          |        |               |        |              |        |          |        |        |        |          |        |        |        |        |        |           |        |           |        |        |
| 001001              | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                 |      |        |                 |        |          |        |          |        |          |        |          |        |               |        |              |        |          |        |        |        |          |        |        |        |        |        |           |        |           |        |        |
| 001010              | LVD_HV                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                 |      |        |                 |        |          |        |          |        |          |        |          |        |               |        |              |        |          |        |        |        |          |        |        |        |        |        |           |        |           |        |        |
| 001011              | HVD_HV                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                 |      |        |                 |        |          |        |          |        |          |        |          |        |               |        |              |        |          |        |        |        |          |        |        |        |        |        |           |        |           |        |        |
| 001100              | LVD_FLASH                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                 |      |        |                 |        |          |        |          |        |          |        |          |        |               |        |              |        |          |        |        |        |          |        |        |        |        |        |           |        |           |        |        |
| 001101              | HVD_FLASH                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |                 |      |        |                 |        |          |        |          |        |          |        |          |        |               |        |              |        |          |        |        |        |          |        |        |        |        |        |           |        |           |        |        |
| 001110              | LVD_IO                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                 |      |        |                 |        |          |        |          |        |          |        |          |        |               |        |              |        |          |        |        |        |          |        |        |        |        |        |           |        |           |        |        |

47.2.8 FCCU Fault Injection Register (PMC\_FIR)

This FCCU Fault Injection register enables user software to artificially generate FCCU faults corresponding to various events, namely LVD, HVD, TEMP SENS and Self Test. A write of 1 to these bits generates a single-cycle pulsed fault to the FCCU block. This bit is automatically cleared to 0 in the next cycle.

Address: 0h base + B8h offset = B8h

|       |    |    |    |    |    |    |    |    |    |    |    |    |             |          |           |           |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|-------------|----------|-----------|-----------|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12          | 13       | 14        | 15        |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |             |          |           |           |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |             |          |           |           |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0           | 0        | 0         | 0         |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28          | 29       | 30        | 31        |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |             | RESERVED |           |           |
| W     |    |    |    |    |    |    |    |    |    |    |    |    | STEST_FAULT |          | HVD_FAULT | LVD_FAULT |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0           | 0        | 0         | 0         |

**PMC\_FIR field descriptions**

| Field             | Description                                                                                                                                                                                              |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–27<br>Reserved  | Reserved<br><br>This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                  |
| 28<br>STEST_FAULT | A write of 1 to this bit sends a FCCU fault (single cycle) for LVD SELF TEST. This is a self clearing bit.<br><br>0 No fault is sent<br>1 Single cycle fault injected.                                   |
| 29<br>RESERVED    | Reserved.<br><br>This field is reserved.<br>Reserved. Read to this bit will return always zero.                                                                                                          |
| 30<br>HVD_FAULT   | High Voltage Detect Fault injection.<br><br>A write of 1 to this bit sends a FCCU fault (single cycle) for HVD. This is a self clearing bit.<br><br>0 No fault is sent<br>1 Single cycle fault injected. |
| 31<br>LVD_FAULT   | Low Voltage Detect Fault injection.<br><br>A write of 1 to this bit sends a FCCU fault (single cycle) for LVD. This is a self clearing bit.<br><br>0 No fault is sent<br>1 Single cycle fault injected.  |

### 47.2.9 LVD/HVD Interrupt Enable Control Register (PMC\_LVD\_HVD\_INTR\_ENB)

This register enables/disables interrupts of PMC VD events.

#### NOTE

Writes to this register must occur in two steps:

1. Set PMC\_ENABLE in this register to 1.
2. In another write operation, program the other fields of the register.

Address: 0h base + BCh offset = BCh

| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

## Memory Map and Registers

| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24       | 25 | 26        | 27       | 28     | 29       | 30       | 31            |
|-------|----|----|----|----|----|----|----|----|----------|----|-----------|----------|--------|----------|----------|---------------|
| R     | 0  |    |    |    |    |    |    |    | Reserved |    | HVD_FLASH | Reserved | HVD_HV | HVD_CORE | Reserved | LVD_CORE_COLD |
| W     |    |    |    |    |    |    |    |    |          |    |           |          |        |          |          |               |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0        | 0  | 0         | 0        | 0      | 0        | 0        | 0             |

### PMC\_LVD\_HVD\_INTR\_ENB field descriptions

| Field             | Description                                                                                                                                                                                                                                                                                                                            |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>PMC_ENABLE   | <p>PMC Enable Control</p> <p>This bit determines whether any of the PMC bits can be written or not.</p> <p>0 No control bit can be written in this(PMC enable control) register.To do a write on this register, first enable this bit and then start writing on other control bits.</p> <p>1 This Control register can be written.</p> |
| 1–23<br>Reserved  | <p>Reserved</p> <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                     |
| 24–25<br>Reserved | <p>Reserved</p> <p>This field is reserved.</p> <p>When writing to this field, always write the reset value.</p>                                                                                                                                                                                                                        |
| 26<br>HVD_FLASH   | <p>HVD Flash Enable/Disable</p> <p>This bit enable/disable the generation of Interrupt due to an event on this supply.</p> <p>0 Interrupt is disabled.</p> <p>1 Interrupt is Enabled.</p>                                                                                                                                              |
| 27<br>Reserved    | <p>Reserved</p> <p>This field is reserved.</p> <p>When writing to this field, always write the reset value.</p>                                                                                                                                                                                                                        |
| 28<br>HVD_HV      | <p>HVD_HV Enable/Disable</p> <p>This bit enable/disable the generation of Interrupt due to an event on this supply.</p> <p>0 Interrupt is disabled.</p> <p>1 Interrupt is Enabled.</p>                                                                                                                                                 |
| 29<br>HVD_CORE    | <p>HVD_CORE Enable/Disable</p> <p>This bit enable/disable the generation of Interrupt due to an event on this supply.</p> <p>0 Interrupt is disabled.</p> <p>1 Interrupt is Enabled.</p>                                                                                                                                               |
| 30<br>Reserved    | <p>Reserved</p> <p>This field is reserved.</p> <p>When writing to this field, always write the reset value.</p>                                                                                                                                                                                                                        |

Table continues on the next page...



**PMC\_LVD\_HVD\_INTR\_ENB field descriptions (continued)**

| Field                   | Description                                                                                                                                                                              |
|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31<br>LVD_CORE_<br>COLD | LVD Core Cold Enable/Disable<br><br>This bit enable/disable the generation of Interrupt due to an event on this supply.<br><br>0    Interrupt is disabled.<br>1    Interrupt is Enabled. |

47.2.10 LVD/HVD Event Status Register  
(PMC\_LVD\_HVD\_EVENT\_STATUS)

This status register contains the latched status of the voltage detect events. These bits indicate the state of the voltage detect events, including whether a voltage level event has occurred at any time in the past but has not been cleared (via writing 1). These bits are read-only, and they are cleared by writing 1 to the corresponding ESR register bit.

Address: 0h base + C0h offset = C0h

|       |    |    |    |    |    |    |    |    |          |          |           |          |        |          |          |               |
|-------|----|----|----|----|----|----|----|----|----------|----------|-----------|----------|--------|----------|----------|---------------|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8        | 9        | 10        | 11       | 12     | 13       | 14       | 15            |
| R     | 0  |    |    |    |    |    |    |    |          |          |           |          |        |          |          |               |
| W     |    |    |    |    |    |    |    |    |          |          |           |          |        |          |          |               |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0        | 0        | 0         | 0        | 0      | 0        | 0        | 0             |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24       | 25       | 26        | 27       | 28     | 29       | 30       | 31            |
| R     | 0  |    |    |    |    |    |    |    | Reserved | Reserved | HVD_FLASH | Reserved | HVD_HV | HVD_CORE | Reserved | LVD_CORE_COLD |
| W     |    |    |    |    |    |    |    |    | w1c      | w1c      | w1c       | w1c      | w1c    | w1c      | w1c      | w1c           |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0        | 0        | 0         | 0        | 0      | 0        | 0        | 0             |

PMC\_LVD\_HVD\_EVENT\_STATUS field descriptions

| Field            | Description                                                                             |
|------------------|-----------------------------------------------------------------------------------------|
| 0–23<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |

Table continues on the next page...

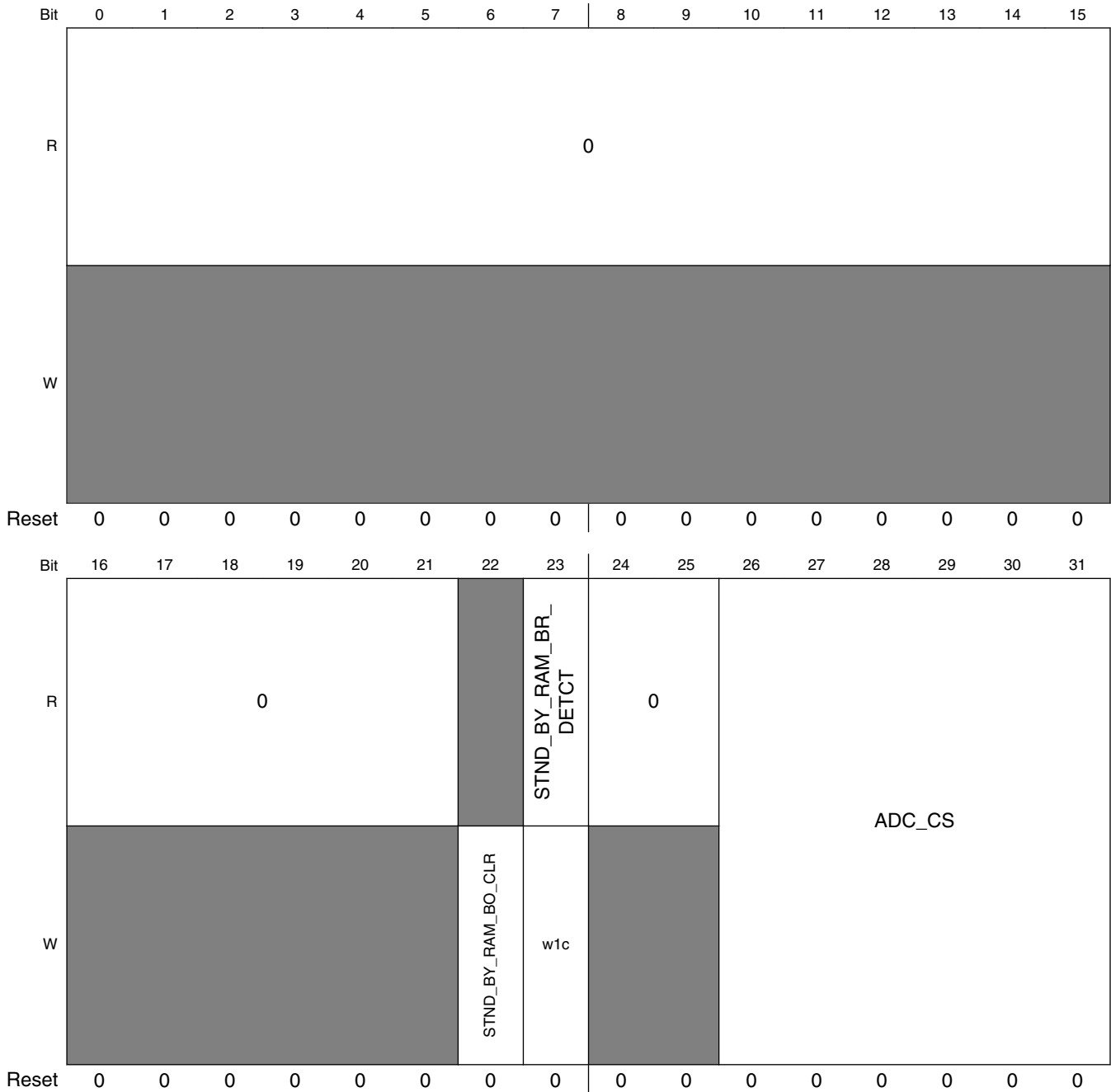
**PMC\_LVD\_HVD\_EVENT\_STATUS field descriptions (continued)**

| <b>Field</b>        | <b>Description</b>                                                                                                                |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| 24<br>Reserved      | Reserved<br><br>This field is reserved.<br>When writing to this field, always write the reset value.                              |
| 25<br>Reserved      | Reserved<br><br>This field is reserved.<br>When writing to this field, always write the reset value.                              |
| 26<br>HVD_FLASH     | This bit controls the enable/disable on the LVD PLL supply.<br><br>0 No occurrence has happened.<br>1 An occurrence has happened. |
| 27<br>Reserved      | Reserved<br><br>This field is reserved.<br>When writing to this field, always write the reset value.                              |
| 28<br>HVD_HV        | HVD_HV Status Flag<br><br>0 No occurrence has happened.<br>1 An occurrence has happened.                                          |
| 29<br>HVD_CORE      | HVD Core Status Flag<br><br>0 No occurrence has happened.<br>1 An occurrence has happened.                                        |
| 30<br>Reserved      | Reserved<br><br>This field is reserved.<br>When writing to this field, always write the reset value.                              |
| 31<br>LVD_CORE_COLD | LVD Core Cold Status Flag<br><br>0 No occurrence has happened.<br>1 An occurrence has happened.                                   |

47.2.11 ADC Channel Select Register (PMC\_ADC\_CS)

This register contains the SRAM Brownout control and the analog channel mux of the analog PMC module.

Address: 0h base + C4h offset = C4h



**PMC\_ADC\_CS field descriptions**

| Field                              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |             |          |        |                 |        |          |        |          |        |                |        |                    |        |          |        |          |        |          |        |          |        |                     |        |                         |        |                    |        |                        |        |                |        |                    |        |                    |        |                                         |        |          |        |          |        |                    |        |                                         |        |                    |
|------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|----------|--------|-----------------|--------|----------|--------|----------|--------|----------------|--------|--------------------|--------|----------|--------|----------|--------|----------|--------|----------|--------|---------------------|--------|-------------------------|--------|--------------------|--------|------------------------|--------|----------------|--------|--------------------|--------|--------------------|--------|-----------------------------------------|--------|----------|--------|----------|--------|--------------------|--------|-----------------------------------------|--------|--------------------|
| 0–21<br>Reserved                   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |             |          |        |                 |        |          |        |          |        |                |        |                    |        |          |        |          |        |          |        |          |        |                     |        |                         |        |                    |        |                        |        |                |        |                    |        |                    |        |                                         |        |          |        |          |        |                    |        |                                         |        |                    |
| 22<br>STND_BY_<br>RAM_BO_CLR       | Standby RAM Brownout Clear<br>A write to this bit generates a 1-clock-wide pulse. It is a self-clearing pulse.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |             |          |        |                 |        |          |        |          |        |                |        |                    |        |          |        |          |        |          |        |          |        |                     |        |                         |        |                    |        |                        |        |                |        |                    |        |                    |        |                                         |        |          |        |          |        |                    |        |                                         |        |                    |
| 23<br>STND_BY_<br>RAM_BR_<br>DETCT | Standby RAM Brownout Detect<br><br>This bit is the detection status flag for the Standby RAM Brownout Detection circuit. When Standby RAM logic detects a brownout event, this bit is set to 1. To clear this flag, two separate write accesses are required: first write 1 to STND_BY_RAM_BO_CLR, and then write 1 to STND_BY_RAM_BR_DETCT.<br><br>0 Currently no occurrence.<br>1 Standby RAM Brownout occurrence detected.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |             |          |        |                 |        |          |        |          |        |                |        |                    |        |          |        |          |        |          |        |          |        |                     |        |                         |        |                    |        |                        |        |                |        |                    |        |                    |        |                                         |        |          |        |          |        |                    |        |                                         |        |                    |
| 24–25<br>Reserved                  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |             |          |        |                 |        |          |        |          |        |                |        |                    |        |          |        |          |        |          |        |          |        |                     |        |                         |        |                    |        |                        |        |                |        |                    |        |                    |        |                                         |        |          |        |          |        |                    |        |                                         |        |                    |
| 26–31<br>ADC_CS                    | Analog-to-Digital Converter Channel Select.<br><br>These bits are used to control the analog PMC's test mux going to the ADC. This is to probe the internal node voltages of the PMC for testing, calibration and characterization purposes by one of the ADCs of the SoC. Encoding values not listed are reserved. <table border="1"> <thead> <tr> <th>ADC_CS[5:0]</th><th>Function</th></tr> </thead> <tbody> <tr><td>000000</td><td>ADC channel off</td></tr> <tr><td>011110</td><td>Reserved</td></tr> <tr><td>011101</td><td>Reserved</td></tr> <tr><td>011100</td><td>POR098_c sense</td></tr> <tr><td>011011</td><td>POR098_c reference</td></tr> <tr><td>011010</td><td>Reserved</td></tr> <tr><td>011001</td><td>Reserved</td></tr> <tr><td>011000</td><td>Reserved</td></tr> <tr><td>010111</td><td>Reserved</td></tr> <tr><td>010110</td><td>LVD_CORE_COLD sense</td></tr> <tr><td>010101</td><td>LVD_CORE_COLD reference</td></tr> <tr><td>010100</td><td>LVD_CORE_HOT sense</td></tr> <tr><td>010011</td><td>LVD_CORE_HOT reference</td></tr> <tr><td>010010</td><td>HVD_CORE sense</td></tr> <tr><td>010001</td><td>HVD_CORE reference</td></tr> <tr><td>010000</td><td>POR_HV divider tap</td></tr> <tr><td>001111</td><td>POR_HV reference (PMC band gap 1200 mV)</td></tr> <tr><td>001110</td><td>Reserved</td></tr> <tr><td>001101</td><td>Reserved</td></tr> <tr><td>001100</td><td>LVD_HV divider tap</td></tr> <tr><td>001011</td><td>LVD_HV reference (PMC band gap 1200 mV)</td></tr> <tr><td>001010</td><td>HVD_HV divider tap</td></tr> </tbody> </table> | ADC_CS[5:0] | Function | 000000 | ADC channel off | 011110 | Reserved | 011101 | Reserved | 011100 | POR098_c sense | 011011 | POR098_c reference | 011010 | Reserved | 011001 | Reserved | 011000 | Reserved | 010111 | Reserved | 010110 | LVD_CORE_COLD sense | 010101 | LVD_CORE_COLD reference | 010100 | LVD_CORE_HOT sense | 010011 | LVD_CORE_HOT reference | 010010 | HVD_CORE sense | 010001 | HVD_CORE reference | 010000 | POR_HV divider tap | 001111 | POR_HV reference (PMC band gap 1200 mV) | 001110 | Reserved | 001101 | Reserved | 001100 | LVD_HV divider tap | 001011 | LVD_HV reference (PMC band gap 1200 mV) | 001010 | HVD_HV divider tap |
| ADC_CS[5:0]                        | Function                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |             |          |        |                 |        |          |        |          |        |                |        |                    |        |          |        |          |        |          |        |          |        |                     |        |                         |        |                    |        |                        |        |                |        |                    |        |                    |        |                                         |        |          |        |          |        |                    |        |                                         |        |                    |
| 000000                             | ADC channel off                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |             |          |        |                 |        |          |        |          |        |                |        |                    |        |          |        |          |        |          |        |          |        |                     |        |                         |        |                    |        |                        |        |                |        |                    |        |                    |        |                                         |        |          |        |          |        |                    |        |                                         |        |                    |
| 011110                             | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |             |          |        |                 |        |          |        |          |        |                |        |                    |        |          |        |          |        |          |        |          |        |                     |        |                         |        |                    |        |                        |        |                |        |                    |        |                    |        |                                         |        |          |        |          |        |                    |        |                                         |        |                    |
| 011101                             | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |             |          |        |                 |        |          |        |          |        |                |        |                    |        |          |        |          |        |          |        |          |        |                     |        |                         |        |                    |        |                        |        |                |        |                    |        |                    |        |                                         |        |          |        |          |        |                    |        |                                         |        |                    |
| 011100                             | POR098_c sense                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |             |          |        |                 |        |          |        |          |        |                |        |                    |        |          |        |          |        |          |        |          |        |                     |        |                         |        |                    |        |                        |        |                |        |                    |        |                    |        |                                         |        |          |        |          |        |                    |        |                                         |        |                    |
| 011011                             | POR098_c reference                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |             |          |        |                 |        |          |        |          |        |                |        |                    |        |          |        |          |        |          |        |          |        |                     |        |                         |        |                    |        |                        |        |                |        |                    |        |                    |        |                                         |        |          |        |          |        |                    |        |                                         |        |                    |
| 011010                             | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |             |          |        |                 |        |          |        |          |        |                |        |                    |        |          |        |          |        |          |        |          |        |                     |        |                         |        |                    |        |                        |        |                |        |                    |        |                    |        |                                         |        |          |        |          |        |                    |        |                                         |        |                    |
| 011001                             | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |             |          |        |                 |        |          |        |          |        |                |        |                    |        |          |        |          |        |          |        |          |        |                     |        |                         |        |                    |        |                        |        |                |        |                    |        |                    |        |                                         |        |          |        |          |        |                    |        |                                         |        |                    |
| 011000                             | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |             |          |        |                 |        |          |        |          |        |                |        |                    |        |          |        |          |        |          |        |          |        |                     |        |                         |        |                    |        |                        |        |                |        |                    |        |                    |        |                                         |        |          |        |          |        |                    |        |                                         |        |                    |
| 010111                             | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |             |          |        |                 |        |          |        |          |        |                |        |                    |        |          |        |          |        |          |        |          |        |                     |        |                         |        |                    |        |                        |        |                |        |                    |        |                    |        |                                         |        |          |        |          |        |                    |        |                                         |        |                    |
| 010110                             | LVD_CORE_COLD sense                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |             |          |        |                 |        |          |        |          |        |                |        |                    |        |          |        |          |        |          |        |          |        |                     |        |                         |        |                    |        |                        |        |                |        |                    |        |                    |        |                                         |        |          |        |          |        |                    |        |                                         |        |                    |
| 010101                             | LVD_CORE_COLD reference                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |             |          |        |                 |        |          |        |          |        |                |        |                    |        |          |        |          |        |          |        |          |        |                     |        |                         |        |                    |        |                        |        |                |        |                    |        |                    |        |                                         |        |          |        |          |        |                    |        |                                         |        |                    |
| 010100                             | LVD_CORE_HOT sense                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |             |          |        |                 |        |          |        |          |        |                |        |                    |        |          |        |          |        |          |        |          |        |                     |        |                         |        |                    |        |                        |        |                |        |                    |        |                    |        |                                         |        |          |        |          |        |                    |        |                                         |        |                    |
| 010011                             | LVD_CORE_HOT reference                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |             |          |        |                 |        |          |        |          |        |                |        |                    |        |          |        |          |        |          |        |          |        |                     |        |                         |        |                    |        |                        |        |                |        |                    |        |                    |        |                                         |        |          |        |          |        |                    |        |                                         |        |                    |
| 010010                             | HVD_CORE sense                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |             |          |        |                 |        |          |        |          |        |                |        |                    |        |          |        |          |        |          |        |          |        |                     |        |                         |        |                    |        |                        |        |                |        |                    |        |                    |        |                                         |        |          |        |          |        |                    |        |                                         |        |                    |
| 010001                             | HVD_CORE reference                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |             |          |        |                 |        |          |        |          |        |                |        |                    |        |          |        |          |        |          |        |          |        |                     |        |                         |        |                    |        |                        |        |                |        |                    |        |                    |        |                                         |        |          |        |          |        |                    |        |                                         |        |                    |
| 010000                             | POR_HV divider tap                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |             |          |        |                 |        |          |        |          |        |                |        |                    |        |          |        |          |        |          |        |          |        |                     |        |                         |        |                    |        |                        |        |                |        |                    |        |                    |        |                                         |        |          |        |          |        |                    |        |                                         |        |                    |
| 001111                             | POR_HV reference (PMC band gap 1200 mV)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |             |          |        |                 |        |          |        |          |        |                |        |                    |        |          |        |          |        |          |        |          |        |                     |        |                         |        |                    |        |                        |        |                |        |                    |        |                    |        |                                         |        |          |        |          |        |                    |        |                                         |        |                    |
| 001110                             | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |             |          |        |                 |        |          |        |          |        |                |        |                    |        |          |        |          |        |          |        |          |        |                     |        |                         |        |                    |        |                        |        |                |        |                    |        |                    |        |                                         |        |          |        |          |        |                    |        |                                         |        |                    |
| 001101                             | Reserved                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |             |          |        |                 |        |          |        |          |        |                |        |                    |        |          |        |          |        |          |        |          |        |                     |        |                         |        |                    |        |                        |        |                |        |                    |        |                    |        |                                         |        |          |        |          |        |                    |        |                                         |        |                    |
| 001100                             | LVD_HV divider tap                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |             |          |        |                 |        |          |        |          |        |                |        |                    |        |          |        |          |        |          |        |          |        |                     |        |                         |        |                    |        |                        |        |                |        |                    |        |                    |        |                                         |        |          |        |          |        |                    |        |                                         |        |                    |
| 001011                             | LVD_HV reference (PMC band gap 1200 mV)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |             |          |        |                 |        |          |        |          |        |                |        |                    |        |          |        |          |        |          |        |          |        |                     |        |                         |        |                    |        |                        |        |                |        |                    |        |                    |        |                                         |        |          |        |          |        |                    |        |                                         |        |                    |
| 001010                             | HVD_HV divider tap                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |             |          |        |                 |        |          |        |          |        |                |        |                    |        |          |        |          |        |          |        |          |        |                     |        |                         |        |                    |        |                        |        |                |        |                    |        |                    |        |                                         |        |          |        |          |        |                    |        |                                         |        |                    |

*Table continues on the next page...*

**PMC\_ADC\_CS field descriptions (continued)**

| Field | Description |                                            |
|-------|-------------|--------------------------------------------|
|       | ADC_CS[5:0] | Function                                   |
|       | 001001      | HVD_HV reference                           |
|       | 001000      | LVD_FLASH divider tap                      |
|       | 000111      | LVD_FLASH reference (PMC band gap 1200 mV) |
|       | 000110      | HVD_FLASH divider tap                      |
|       | 000101      | HVD_FLASH reference                        |
|       | 000100      | LVD_IO divider tap                         |
|       | 000011      | LVD_IO reference                           |

**47.2.12 Reset Event Enable Register (PMC\_REE)**

This Reset Event Enable Register controls whether or not the voltage detect signal event causes a reset. If the desired flag bit is enabled, the voltage detect event causes a POR to be generated when the selected voltage passes the trigger event. These bits are loaded from the flash memory during the boot sequence. If the bits loaded from flash memory are programmed to be enabled (set), these bits cannot be disabled (cleared).

**NOTE**

This register is cleared on POR or a reset due to RESET pin assertion or FOSU failure indication.

**CAUTION**

Except when they occur during a PMC self test, the following VD events *always* result in a POR. The result of these VD events cannot be programmed:

- POR098\_c
- LVD\_CORE\_HOT
- POR\_HV
- LVD\_HV
- LVD\_FLASH
- LVD\_IO

Address: 0h base + D0h offset = D0h

|       |    |    |    |    |    |    |    |    |          |    |           |          |        |          |          |               |
|-------|----|----|----|----|----|----|----|----|----------|----|-----------|----------|--------|----------|----------|---------------|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8        | 9  | 10        | 11       | 12     | 13       | 14       | 15            |
| R     | 0  |    |    |    |    |    |    |    |          |    |           |          |        |          |          |               |
| W     |    |    |    |    |    |    |    |    |          |    |           |          |        |          |          |               |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0        | 0  | 0         | 0        | 0      | 0        | 0        | 0             |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24       | 25 | 26        | 27       | 28     | 29       | 30       | 31            |
| R     | 0  |    |    |    |    |    |    |    | Reserved |    | HVD_FLASH | Reserved | HVD_HV | HVD_CORE | Reserved | LVD_CORE_COLD |
| W     |    |    |    |    |    |    |    |    |          |    |           |          |        |          |          |               |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 1        | 1  | 1         | 1        | 1      | 1        | 1        | 1             |

**PMC\_REE field descriptions**

| Field             | Description                                                                                                                                                                                                                                                                                                   |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–23<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                       |
| 24–25<br>Reserved | Reserved<br><br>This field is reserved.<br>When writing to this field, always write the reset value.                                                                                                                                                                                                          |
| 26<br>HVD_FLASH   | Reset Enable/Disable Control<br><br>This particular bit enable or disable the generation of reset due to an event on this supply.<br><br>0 Reset is disabled, no reset will be generated even if this event occurred.<br>1 Reset is enabled. A reset will be generated if an event is detected on this supply |
| 27<br>Reserved    | Reserved<br><br>This field is reserved.<br>When writing to this field, always write the reset value.                                                                                                                                                                                                          |
| 28<br>HVD_HV      | Reset Enable/Disable Control<br><br>This particular bit enable or disable the generation of reset due to an event on this supply.<br><br>0 Reset is disabled, no reset will be generated even if this event occurred.<br>1 Reset is enabled. A reset will be generated if an event is detected on this supply |
| 29<br>HVD_CORE    | Reset Enable/Disable Control<br><br>This particular bit enable or disable the generation of reset due to an event on this supply.<br><br>0 Reset is disabled, no reset will be generated even if this event occurred.<br>1 Reset is enabled. A reset will be generated if an event is detected on this supply |
| 30<br>Reserved    | Reserved<br><br>This field is reserved.                                                                                                                                                                                                                                                                       |

*Table continues on the next page...*

**PMC\_REE field descriptions (continued)**

| Field                   | Description                                                                                                                                                                                                                                                                                                                           |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                         | When writing to this field, always write the reset value.                                                                                                                                                                                                                                                                             |
| 31<br>LVD_CORE_<br>COLD | Reset Enable/Disable Control<br><br>This particular bit enable or disable the generation of reset due to an event on this supply.<br><br>0   Reset is disabled, no reset will be generated even if there is event occurred on this supply.<br>1   Reset is enabled. A reset will be generated if an event is detected on this supply. |

## 47.3 Functional description

### 47.3.1 Analog PMC interface

The digital block uses the analog PMC interface to configure factory-defined trim values for the regulators and LVDs that are in the analog block.

### 47.3.2 Temperature Sensor interface logic

The digital PMC block provides control and configuration information, including module enables, to the two Temperature Sensors.

### 47.3.3 POR phase gates

The interface with the SIU module is used during the power-on sequence and startup. It provides various phase gates, thereby enabling the SIU to properly complete the reset sequence.

It also provides reset sources based on LVD and HVD events and their enable/disable control through the REE register.

The temperature sensor provides another reset source, based on the under/over temperature events and their control through the REE\_TD and RES\_TD registers.



### 47.3.4 Flash memory interface to DCF

During power up, trim values for the regulators, LVDs, and HVDs are read from flash memory by the SSCM and then provided to the digital PMC via the DCF interface.

The reset event enable (REE) configurations that are stored in flash memory are also provided to the PMC via the DCF interface. These REE configurations are used, by default, to configure which programmable voltage detection (VD) events generate a POR.

### 47.3.5 FCCU interface

The digital block generates various fault indications to the FCCU block:

- Separate, corresponding, active-high fault indications are generated whenever an LVD or HVD defect is detected in the analog block.
- A fault indication is generated when a self test task results in a failure.
- A fault condition is generated whenever over-temperature or under-temperature events occur in the temperature sensor.

The figure below show the exact logic for the generation fault indications to FCCU block.

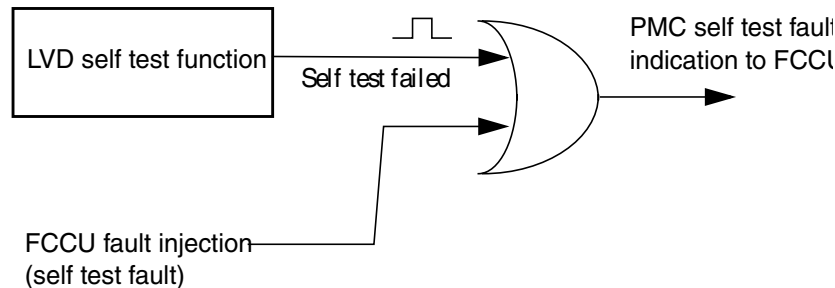


**Figure 47-13. Fault generation to FCCU**

See [FCCU Fault Injection Register \(PMC\\_FIR\)](#) for details.

### 47.3.6 LVD/HVD self test mechanism

The PMC implements an LVD/HVD self test mechanism for use in satisfying safety goals. The digital block assists and controls the analog block during execution of the self test.

Currently, three self test modes are supported by the analog block. Select the mode using the SELF\_TEST\_UM\_VD\_REG[ST\_MODE] field.

The self test status and result are provided in the SELF\_TEST\_UM\_VD\_REG register's ST\_DONE and ST\_RESULT fields, respectively.

### **NOTE**

Self tests must execute with the FM\_PER\_CLK frequency at 8 MHz.

#### **47.3.6.1 Default mode**

The self test always occurs during startup (power on). The self test starts after the SSCM has completed reads from the flash memory. A counter is provided that starts from zero and increments by one after a factory-defined time gap. The counter value is the source for the self test output signal, which is provided to the analog block.

Meanwhile, the LVD/HVD indication from the analog block is latched in the SELF\_TEST\_UM\_VD\_REG register. At the end of the self test, after all of the LVDs/HVDs have been tested, the SELF\_TEST\_UM\_VD\_REG[ST\_DONE] status bit is set to 1 and all bits in the SELF\_TEST\_UM\_VD\_REG register are read to see if the value of each is 1.

- If one or more bits is not set to 1, a fault indication to the FCCU block is generated, and the SELF\_TEST\_UM\_VD\_REG[ST\_RESULT] field is set to 0.
- If each of the register bits is set to 1, the test is considered passed, the SELF\_TEST\_UM\_VD\_REG[ST\_RESULT] bit is set to 1, and no fault indications are provided to the FCCU.

### **NOTE**

The total run time of the self test is 300  $\mu$ s (factory-defined default).

#### **47.3.6.2 Software-triggered self test**

This mode is entered by software by writing 01b to the SELF\_TEST\_UM\_VD\_REG[ST\_MODE] field. The complete self test, as is performed in default mode, is executed. The difference is that the self test in this mode is initiated by the host while the self test in default mode is started automatically during power on.

### 47.3.6.3 Single voltage detect test

This mode is entered by software writing 10b to the SELF\_TEST\_UM\_VD\_REG[ST\_MODE] field. The test completes after the prescribed time gap has elapsed (factory-defined default is 20  $\mu$ s). The PASS/FAIL result is then updated based on whether or not the corresponding VD output trips.

#### NOTE

During execution of the VD self test, all of the LVD and HVD outputs—including PORs, which are also under self test—are masked. Therefore, they have no effect on POR generation or any reset and/or fault generation.



# Chapter 48

## JTAG Controller (JTAGC)

### 48.1 Chip-specific JTAGC information

This chip has one instance of the JTAGC module.

#### 48.1.1 JTAG ID

The JTAG ID depends on the version of the chip.

**Table 48-1. JTAG ID by chip version**

| Maskset | JTAG ID    |
|---------|------------|
| 3N45H   | 1837_701Dh |

For details, see [Device identification register](#).

### 48.2 Introduction

The JTAGC block provides the means to test chip functionality and connectivity while remaining transparent to system logic when not in test mode. Testing is performed via a boundary scan technique, as defined in the IEEE 1149.1-2001 standard. All data input to and output from the JTAGC block is communicated in serial format.

#### 48.2.1 Block diagram

The following is a simplified block diagram of the JTAG Controller (JTAGC) block. Refer to [Register description](#) for more information about the JTAGC registers.

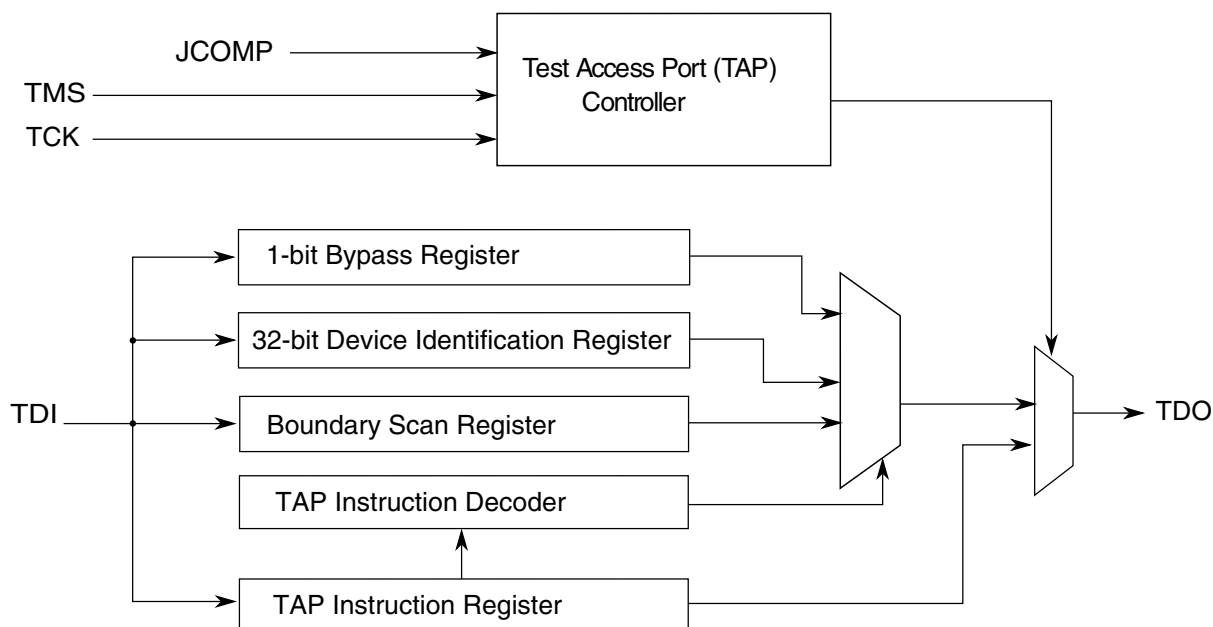


**Figure 48-1. JTAG (IEEE 1149.1) block diagram**

## 48.2.2 Features

The JTAGC block is compliant with the IEEE 1149.1-2001 standard, and supports the following features:

- IEEE 1149.1-2001 Test Access Port (TAP) interface
  - 4 pins (TDI, TMS, TCK, and TDO)
- JCOMP input that provides reset control
- Instruction register that supports several IEEE 1149.1-2001 defined instructions as well as several public and private device-specific instructions. Refer to [Table 48-5](#) for a list of supported instructions.
- Sharing of the TAP with other TAP controllers via ACCESS\_AUX\_x instructions
- JTAG\_PASSWORD register
- Bypass register, boundary scan register, and device identification register.
- TAP controller state machine that controls the operation of the data registers, instruction register and associated circuitry.

## 48.2.3 Modes of operation

The JTAGC block uses JCOMP and a power-on reset indication as its primary reset signals. Several IEEE 1149.1-2001 defined test modes are supported, as well as a bypass mode.

### 48.2.3.1 Reset

The JTAGC block is placed in reset when either power-on reset is asserted, JCOMP is negated, or the TMS input is held high for enough consecutive rising edges of TCK to sequence the TAP controller state machine into the Test-Logic-Reset state. Holding TMS high for five consecutive rising edges of TCK guarantees entry into the Test-Logic-Reset state regardless of the current TAP controller state. Asserting power-on reset or setting JCOMP to a value other than the value required to enable the JTAGC block results in asynchronous entry into the reset state. While in reset, the following actions occur:

- The TAP controller is forced into the Test-Logic-Reset state, thereby disabling the test logic and allowing normal operation of the on-chip system logic to continue unhindered
- The instruction register is loaded with the IDCODE instruction

### 48.2.3.2 IEEE 1149.1-2001 defined test modes

The JTAGC block supports several IEEE 1149.1-2001 defined test modes. A test mode is selected by loading the appropriate instruction into the instruction register while the JTAGC is enabled. Supported test instructions include EXTEST, HIGHZ, CLAMP, SAMPLE and SAMPLE/PRELOAD. Each instruction defines the set of data register(s) that may operate and interact with the on-chip system logic while the instruction is current. Only one test data register path is enabled to shift data between TDI and TDO for each instruction.

The boundary scan register is enabled for serial access between TDI and TDO when the EXTEST, SAMPLE or SAMPLE/PRELOAD instructions are active. The single-bit bypass register shift stage is enabled for serial access between TDI and TDO when the BYPASS, HIGHZ, CLAMP or reserved instructions are active. The functionality of each test mode is explained in more detail in [JTAGC block instructions](#).

### 48.2.3.3 Bypass mode

When no test operation is required, the BYPASS instruction can be loaded to place the JTAGC block into bypass mode. While in bypass mode, the single-bit bypass shift register is used to provide a minimum-length serial path to shift data between TDI and TDO.

## 48.3 External signal description

The JTAGC consists of a set of signals that connect to off chip development tools and allow access to test support functions. The JTAGC signals are outlined in the following table and described in the following sections.

**Table 48-2. JTAG signal properties**

| Name  | I/O    | Function         | Reset State         |
|-------|--------|------------------|---------------------|
| TCK   | Input  | Test Clock       | Weak pulldown       |
| TDI   | Input  | Test Data In     | Weak pullup         |
| TDO   | Output | Test Data Out    | High Z <sup>1</sup> |
| TMS   | Input  | Test Mode Select | Weak pullup         |
| JCOMP | Input  | JTAG Compliancy  | Weak pulldown       |

1. TDO output buffer enable is negated when the JTAGC is not in the Shift-IR or Shift-DR states. A weak pull may be implemented at the TDO pad for use when JTAGC is inactive.

### 48.3.1 TCK—Test clock input

Test Clock Input (TCK) is an input pin used to synchronize the test logic and control register access through the TAP.

### 48.3.2 TDI—Test data input

Test Data Input (TDI) is an input pin that receives serial test instructions and data. TDI is sampled on the rising edge of TCK.



### 48.3.3 TDO—Test data output

Test Data Output (TDO) is an output pin that transmits serial output for test instructions and data. TDO is three-stateable and is actively driven only in the Shift-IR and Shift-DR states of the TAP controller state machine, which is described in [TAP controller state machine](#).

### 48.3.4 TMS—Test mode select

Test Mode Select (TMS) is an input pin used to sequence the IEEE 1149.1-2001 test control state machine. TMS is sampled on the rising edge of TCK.

### 48.3.5 JCOMP—JTAG compliancy

The JCOMP signal provides IEEE 1149.1-2001 compatibility and provides the ability to share the TAP. The JTAGC TAP controller is enabled when JCOMP is set to the JTAGC enable encoding, otherwise the JTAGC TAP controller remains in reset.

## 48.4 Register description

This section provides a detailed description of the JTAGC block registers accessible through the TAP interface, including data registers and the instruction register. Individual bit-level descriptions and reset states of each register are included. These registers are not memory-mapped and can only be accessed through the TAP.

### 48.4.1 Instruction register

The JTAGC block uses a 5-bit instruction register as shown in the following figure. The instruction register allows instructions to be loaded into the block to select the test to be performed or the test data register to be accessed or both. Instructions are shifted in through TDI while the TAP controller is in the Shift-IR state, and latched on the falling edge of TCK in the Update-IR state. The latched instruction value can only be changed in the Update-IR and Test-Logic-Reset TAP controller states. Synchronous entry into the Test-Logic-Reset state results in the IDCODE instruction being loaded on the falling edge of TCK. Asynchronous entry into the Test-Logic-Reset state results in asynchronous loading of the IDCODE instruction. During the Capture-IR TAP controller state, the instruction shift register is loaded with the value 00001b, making this value the register's read value when the TAP controller is sequenced into the Shift-IR state.

|        |                  |   |   |   |   |
|--------|------------------|---|---|---|---|
|        | 4                | 3 | 2 | 1 | 0 |
| R      | 0                | 0 | 0 | 0 | 1 |
| W      | Instruction Code |   |   |   |   |
| Reset: | 0                | 0 | 0 | 0 | 1 |

Figure 48-2. Instruction register

### 48.4.2 Bypass register

The bypass register is a single-bit shift register path selected for serial data transfer between TDI and TDO when the BYPASS, CLAMP, HIGHZ or reserve instructions are active. After entry into the Capture-DR state, the single-bit shift register is set to a logic 0. Therefore, the first bit shifted out after selecting the bypass register is always a logic 0.

### 48.4.3 Device identification register

The device identification (JTAG ID) register, shown in the following figure, allows the revision number, part number, manufacturer, and design center responsible for the design of the part to be determined through the TAP. The device identification register is selected for serial data transfer between TDI and TDO when the IDCODE instruction is active. Entry into the Capture-DR state while the device identification register is selected loads the IDCODE into the shift register to be shifted out on TDO in the Shift-DR state. No action occurs in the Update-DR state.

|       |                            |    |    |    |    |    |    |    |                            |    |    |    |    |    |                            |    |
|-------|----------------------------|----|----|----|----|----|----|----|----------------------------|----|----|----|----|----|----------------------------|----|
|       | 31                         | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23                         | 22 | 21 | 20 | 19 | 18 | 17                         | 16 |
| R     | Part Revision Number       |    |    |    |    |    |    |    | Design Center              |    |    |    |    |    | Part Identification Number |    |
| W     |                            |    |    |    |    |    |    |    |                            |    |    |    |    |    |                            |    |
| Reset | PRN                        |    |    |    |    |    |    |    | DC                         |    |    |    |    |    | PIN                        |    |
|       | 15                         | 14 | 13 | 12 | 11 | 10 | 9  | 8  | 7                          | 6  | 5  | 4  | 3  | 2  | 1                          | 0  |
| R     | Part Identification Number |    |    |    |    |    |    |    | Manufacturer Identity Code |    |    |    |    |    |                            | 1  |
| W     |                            |    |    |    |    |    |    |    |                            |    |    |    |    |    |                            |    |
| Reset | PIN (contd.)               |    |    |    |    |    |    |    | MIC                        |    |    |    |    |    |                            | 1  |

The following table describes the device identification register functions.

Table 48-3. Device identification register field descriptions

| Field | Description                                                                                                                       |
|-------|-----------------------------------------------------------------------------------------------------------------------------------|
| PRN   | Part Revision Number. Contains the revision number of the part. Value is 0x0 for masksets 1N45H and 2N45H, 0x1 for maskset 3N45H. |
| DC    | Design Center. Indicates the design center. Value is 0x20.                                                                        |
| PIN   | Part Identification Number. Contains the part number of the device. 0x377.                                                        |

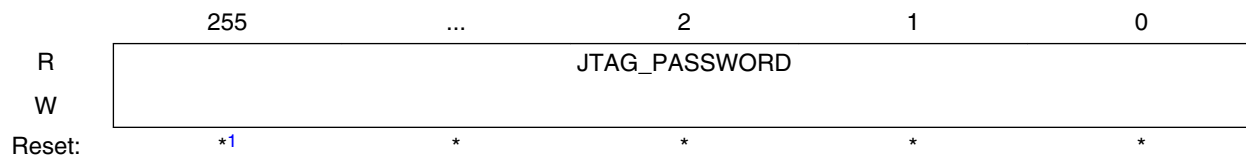
Table continues on the next page...

**Table 48-3. Device identification register field descriptions (continued)**

| Field     | Description                                                                                                                      |
|-----------|----------------------------------------------------------------------------------------------------------------------------------|
| MIC       | Manufacturer Identity Code. Contains the reduced Joint Electron Device Engineering Council (JEDEC) ID. Value is 0x00E.           |
| IDCODE ID | IDCODE Register ID. Identifies this register as the device identification register and not the bypass register. Always set to 1. |

#### 48.4.4 JTAG\_PASSWORD register

The JTAG\_PASSWORD register is a 256-bit shift register path from TDI to TDO selected when the ENABLE\_JTAG\_PASSWORD instruction is active. The default reset value of the JTAG\_PASSWORD register is 256'b0. The JTAG\_PASSWORD register transfers its value to a parallel hold register on the rising edge of TCK when the TAP controller state machine is in the Update-DR state. Once the ENABLE\_JTAG\_PASSWORD instruction is executed, the register value remains valid until a JTAG reset occurs. The operation of this register is described in the security documentation.



1. The reset value of JTAG\_PASSWORD is 256 'b0.

The following table describes the JTAG\_PASSWORD register functions.

**Table 48-4. JTAG\_PASSWORD register field descriptions**

| Field         | Description                                                                               |
|---------------|-------------------------------------------------------------------------------------------|
| JTAG_PASSWORD | JTAG Password. The JTAG_PASSWORD bits are used to provide the JTAG password for security. |

#### 48.4.5 Boundary scan register

The boundary scan register is connected between TDI and TDO when the EXTEST, SAMPLE or SAMPLE/PRELOAD instructions are active. It is used to capture input pin data, force fixed values on output pins, and select a logic value and direction for bidirectional pins. Each bit of the boundary scan register represents a separate boundary

scan register cell, as described in the IEEE 1149.1-2001 standard and discussed in [Boundary scan](#). The size of the boundary scan register and bit ordering is device-dependent and can be found in the device BSDL file.

## 48.5 Functional description

This section explains the JTAGC functional description.

### 48.5.1 JTAGC reset configuration

While in reset, the TAP controller is forced into the Test-Logic-Reset state, thus disabling the test logic and allowing normal operation of the on-chip system logic. In addition, the instruction register is loaded with the IDCODE instruction.

### 48.5.2 IEEE 1149.1-2001 (JTAG) Test Access Port

The JTAGC block uses the IEEE 1149.1-2001 TAP for accessing registers. This port can be shared with other TAP controllers on the MCU. Ownership of the port is determined by the value of the currently loaded instruction. For more detail on TAP sharing via JTAGC instructions refer to [ACCESS\\_AUX\\_x instructions](#).

Data is shifted between TDI and TDO through the selected register starting with the least significant bit, as illustrated in the following figure. This applies for the instruction register, test data registers, and the bypass register.

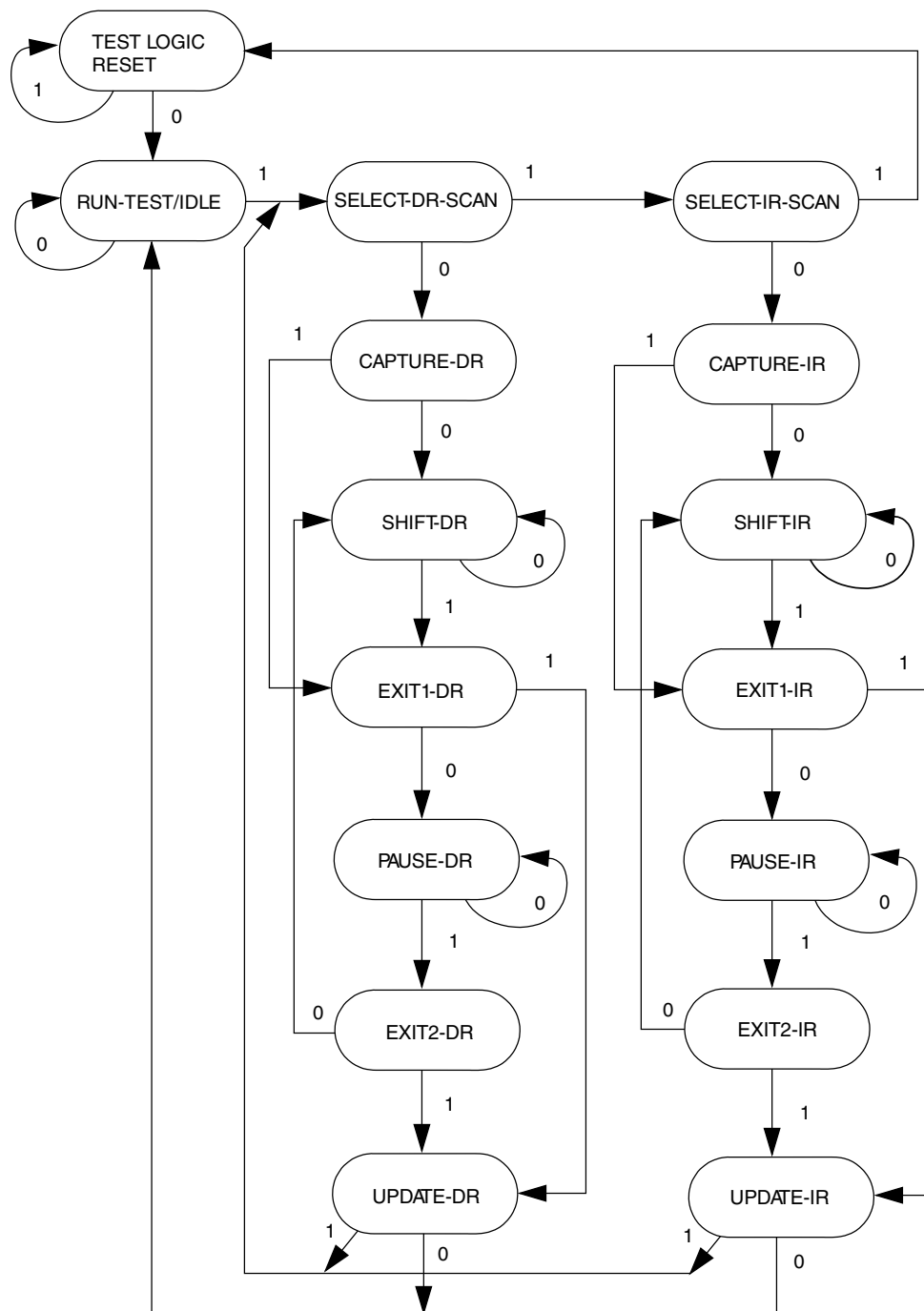


**Figure 48-3. Shifting data through a register**

### 48.5.3 TAP controller state machine

The TAP controller is a synchronous state machine that interprets the sequence of logical values on the TMS pin. The following figure shows the machine's states. The value shown next to each state is the value of the TMS signal sampled on the rising edge of the

TCK signal. As the following figure shows, holding TMS at logic 1 while clocking TCK through a sufficient number of rising edges also causes the state machine to enter the Test-Logic-Reset state.



The value shown adjacent to each state transition in this figure represents the value of TMS at the time of a rising edge of TCK.

**Figure 48-4. IEEE 1149.1-2001 TAP controller finite state machine**

### 48.5.3.1 Enabling the TAP controller

The JTAGC TAP controller is enabled by setting JCOMP to a logic 1 value.

### 48.5.3.2 Selecting an IEEE 1149.1-2001 register

Access to the JTAGC data registers is achieved by loading the instruction register with any of the JTAGC block instructions while the JTAGC is enabled. Instructions are shifted in via the Select-IR-Scan path and loaded in the Update-IR state. At this point, all data register access is performed via the Select-DR-Scan path.

The Select-DR-Scan path is used to read or write the register data by shifting in the data (LSB first) during the Shift-DR state. When reading a register, the register value is loaded into the IEEE 1149.1-2001 shifter during the Capture-DR state. When writing a register, the value is loaded from the IEEE 1149.1-2001 shifter to the register during the Update-DR state. When reading a register, there is no requirement to shift out the entire register contents. Shifting may be terminated once the required number of bits have been acquired.

## 48.5.4 JTAGC block instructions

The JTAGC block implements the IEEE 1149.1-2001 defined instructions listed in the following table. This section gives an overview of each instruction; refer to the IEEE 1149.1-2001 standard for more details. All undefined opcodes are reserved.

**Table 48-5. General 5-bit JTAG instructions**

| Instruction            | Code[4:0] | Instruction Summary                                                                                                                                     |
|------------------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| IDCODE                 | 00001     | Selects device identification register for shift                                                                                                        |
| SAMPLE/PRELOAD         | 00010     | Selects boundary scan register for shifting, sampling, and preloading without disturbing functional operation                                           |
| SAMPLE                 | 00011     | Selects boundary scan register for shifting and sampling without disturbing functional operation                                                        |
| EXTEST                 | 00100     | Selects boundary scan register and applies preloaded values to output pins.<br><br><b>NOTE:</b> Execution of this instruction asserts functional reset. |
| Factory debug reserved | 00101     | Intended for factory debug only                                                                                                                         |
| Factory debug reserved | 00110     | Intended for factory debug only                                                                                                                         |
| ENABLE_JTAG_PASSWORD   | 00111     | Selects JTAG_PASSWORD register                                                                                                                          |
| HIGHZ                  | 01001     | Selects bypass register and three-states all output pins.<br><br><b>NOTE:</b> Execution of this instruction asserts functional reset.                   |
| Factory debug reserved | 01010     | Intended for factory debug only                                                                                                                         |

*Table continues on the next page...*

**Table 48-5. General 5-bit JTAG instructions (continued)**

| Instruction            | Code[4:0]         | Instruction Summary                                                                                                                          |
|------------------------|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| CLAMP                  | 01100             | Selects bypass register and applies preloaded values to output pins.<br><b>NOTE:</b> Execution of this instruction asserts functional reset. |
| Factory debug reserved | 01110             | Intended for factory debug only                                                                                                              |
| Factory debug reserved | 01111             | Intended for factory debug only                                                                                                              |
| BYPASS                 | 11111             | Selects bypass register for data operations                                                                                                  |
| Reserved <sup>1</sup>  | All other opcodes | Decoded to select bypass register                                                                                                            |

1. The manufacturer reserves the right to change the decoding of reserved instruction codes in the future

The ACCESS\_AUX instructions are described in the chip configuration debug information.

#### 48.5.4.1 IDCODE instruction

IDCODE selects the 32-bit device identification register as the shift path between TDI and TDO. This instruction allows interrogation of the MCU to determine its version number and other part identification data. IDCODE is the instruction placed into the instruction register when the JTAGC block is reset.

#### 48.5.4.2 SAMPLE/PRELOAD instruction

The SAMPLE/PRELOAD instruction has two functions:

- The SAMPLE portion of the instruction obtains a sample of the system data and control signals present at the MCU input pins and just before the boundary scan register cells at the output pins. This sampling occurs on the rising edge of TCK in the Capture-DR state when the SAMPLE/PRELOAD instruction is active. The sampled data is viewed by shifting it through the boundary scan register to the TDO output during the Shift-DR state. Both the data capture and the shift operation are transparent to system operation.
- The PRELOAD portion of the instruction initializes the boundary scan register cells before selecting the EXTEST or CLAMP instructions to perform boundary scan tests. This is achieved by shifting in initialization data to the boundary scan register during the Shift-DR state. The initialization data is transferred to the parallel outputs of the boundary scan register cells on the falling edge of TCK in the Update-DR state. The data is applied to the external output pins by the EXTEST or CLAMP instruction. System operation is not affected.

### 48.5.4.3 SAMPLE instruction

The SAMPLE instruction obtains a sample of the system data and control signals present at the MCU input pins and just before the boundary scan register cells at the output pins. This sampling occurs on the rising edge of TCK in the Capture-DR state when the SAMPLE instruction is active. The sampled data is viewed by shifting it through the boundary scan register to the TDO output during the Shift-DR state. There is no defined action in the Update-DR state. Both the data capture and the shift operation are transparent to system operation.

### 48.5.4.4 EXTEST External test instruction

EXTEST selects the boundary scan register as the shift path between TDI and TDO. It allows testing of off-chip circuitry and board-level interconnections by driving preloaded data contained in the boundary scan register onto the system output pins. Typically, the preloaded data is loaded into the boundary scan register using the SAMPLE/PRELOAD instruction before the selection of EXTEST. EXTEST asserts the internal system reset for the MCU to force a predictable internal state while performing external boundary scan operations.

### 48.5.4.5 ENABLE\_JTAG\_PASSWORD instruction

The ENABLE\_JTAG\_PASSWORD instruction selects the JTAG\_PASSWORD register for connection as the shift path between TDI and TDO.

### 48.5.4.6 HIGHZ instruction

HIGHZ selects the bypass register as the shift path between TDI and TDO. While HIGHZ is active all output drivers are placed in an inactive drive state (e.g., high impedance). HIGHZ also asserts the internal system reset for the MCU to force a predictable internal state.

### 48.5.4.7 CLAMP instruction

CLAMP allows the state of signals driven from MCU pins to be determined from the boundary scan register while the bypass register is selected as the serial path between TDI and TDO. CLAMP enhances test efficiency by reducing the overall shift path to a



single bit (the bypass register) while conducting an EXTEST type of instruction through the boundary scan register. CLAMP also asserts the internal system reset for the MCU to force a predictable internal state.

#### **48.5.4.8 ACCESS\_AUX\_x instructions**

The JTAGC is configurable to allow other TAP controllers on the device to share the port with it. This is done by providing ACCESS\_AUX\_x instructions for each of these TAP controllers. When this instruction is loaded, control of the JTAG pins are transferred to the selected TAP controller. Any data input via TDI and TMS is passed to the selected TAP controller, and any TDO output from the selected TAP controller is sent back to the JTAGC to be output on the pins. The JTAGC regains control of the JTAG port during the UPDATE-DR state if the PAUSE-DR state was entered. Auxiliary TAP controllers are held in RUN-TEST/IDLE while they are inactive. Instructions not used to access an auxiliary TAP controller on a device are treated like the BYPASS instruction.

#### **48.5.4.9 BYPASS instruction**

BYPASS selects the bypass register, creating a single-bit shift register path between TDI and TDO. BYPASS enhances test efficiency by reducing the overall shift path when no test operation of the MCU is required. This allows more rapid movement of test data to and from other components on a board that are required to perform test functions. While the BYPASS instruction is active the system logic operates normally.

### **48.5.5 Boundary scan**

The boundary scan technique allows signals at component boundaries to be controlled and observed through the shift-register stage associated with each pad. Each stage is part of a larger boundary scan register cell, and cells for each pad are interconnected serially to form a shift-register chain around the border of the design. The boundary scan register consists of this shift-register chain, and is connected between TDI and TDO when the EXTEST, SAMPLE, or SAMPLE/PRELOAD instructions are loaded. The shift-register chain contains a serial input and serial output, as well as clock and control signals.

## 48.6 Initialization/Application information

The test logic is a static logic design, and TCK can be stopped in either a high or low state without loss of data. However, the system clock is not synchronized to TCK internally. Any mixed operation using both the test logic and the system functional logic requires external synchronization.

To initialize the JTAGC block and enable access to registers, the following sequence is required:

1. Set the JCOMP signal to the JTAGC enable value, thereby enabling the JTAGC TAP controller
2. Load the appropriate instruction for the test or action to be performed

# Chapter 49

## IEEE 1149.7 Compact JTAG Test Access Port Controller (CJTAG)

### 49.1 References

- IEEE Std 1149.1-2001 (R2008), Standard Test Access Port and Boundary-Scan Architecture
- IEEE Std 1149.7, Standard for Reduced-Pin and Enhanced-Functionality Test Access Port and Boundary-Scan Architecture

### 49.2 Abbreviations

**Table 49-1. Abbreviations**

| Abbreviation | Expansion                          |
|--------------|------------------------------------|
| APU          | Advanced Protocol Unit             |
| APFC         | Auxiliary Pin Function Control     |
| CGM          | Conditional Group Member           |
| CGMC         | Conditional Group Membership Count |
| CID          | Controller ID                      |
| CJTAG        | Compact JTAG                       |
| CLTAPC       | Chip-Level TAPC                    |
| CP           | Check Packet                       |
| CPA          | Check Process Active               |
| DCU          | Data Channel Unit                  |
| EOT          | End of Transfer                    |
| EPU          | Extended Protocol Unit             |
| Jscan        | JTAG Scan                          |
| Mscan        | Maximum-flexibility Scan           |
| nTRST        | Active low Test Reset              |
| Oscan        | Optimized Scan                     |
| PSS          | Pause Selection State              |

*Table continues on the next page...*

**Table 49-1. Abbreviations (continued)**

| Abbreviation | Expansion                       |
|--------------|---------------------------------|
| RSU          | Reset and Selection Unit        |
| SP           | Scan Packet                     |
| SPA          | Scan Packet Active              |
| Sscan        | Segmented Scan                  |
| SSD          | Scan Selection Directive        |
| SSM          | Scan State Machine              |
| STL          | System Test Logic               |
| TAP          | Test Access Port                |
| TAP.1        | An IEEE 1149.1 Test Access Port |
| TAP.7        | An IEEE 1149.7 Test Access Port |
| TCA          | Tap Controller Address          |
| TCK          | Test Clock                      |
| TCKC         | Test Clock Compact              |
| TDI          | Test Data Input                 |
| TDIC         | Test Data Input Compact         |
| TDO          | Test Data Output                |
| TDOC         | Test Data Output Compact        |
| TMS          | Test Mode Select                |
| TMSC         | Test Mode Select Compact        |
| TS           | Target System                   |
| ZBS          | Zero Bit Scan                   |

## 49.3 Introduction

The CJTAG module implements parts of the IEEE 1149.7 standard for test and debug capabilities. IEEE 1149.7 defines enhanced test and debug capabilities which are backwards compatible with IEEE P1149.1. This standard is also known as "Compact JTAG" or "CJTAG".

The module implements the following parts of the IEEE 1149.7 standard:

- IEEE 1149.1 compliant behavior while allowing multiple on-chip TAP controllers
- Ability to assert test reset
- Improved system scan performance with one-bit IR and DR chip bypass paths

- Physical connection of a Debug and Test System (external tool) to a running system without corrupting its operation
- Operation with both 4-pin and 2-pin scan topologies

### 49.3.1 Types of operation

The 1149.7 standard provides a scalable and flexible solution to meet the needs of varied component and system complexities. There are three types of operation:

- **Compliant**
  - Compliant with the IEEE 1149.1 standard
  - Four-pin operation using the "Standard Protocol" — IEEE 1149.1 signaling where TCK, TMS, TDI and TDO signals transfer control/data information
  - More than one TAPC may be associated with the TAP (once made visible by private instructions)
- **Extended**
  - Compatible with the IEEE 1149.1 standard
  - Four-pin operation using the "Standard Protocol" — IEEE 1149.1 signaling where TCK, TMS, TDI(C) and TDO(C) signals transfer control/data information
  - Additional test/debug functions
- **Advanced**
  - Two-pin operation using the "Advanced Protocol" — signaling where TCK(C) and TMS(C) signals transfer control/data information
  - The TMS(C) signal is operated in a bidirectional manner with the TMS, TDI and TDO TAP information serialized and transferred via this signal

### 49.3.2 Deployment by class

The TAP.7 capabilities are deployed using five capability classes (T0–T4), each matching the needs of certain scan topologies and mixes of 1149.1 TAPs and 1149.7 TAPs. The classes provide the following capability:

- **Compliant Operation** – provides behavior compliant with the IEEE 1149.1-2001 standard:
  - **Class T0** – 1149.1 compliance when multiple TAPs are used on the same chip.
- **Extended Operation** – extends the capabilities of the IEEE 1149.1-2001 standard:
  - **Class T1** – T0 + Test access port power management, functional reset, and test reset.
  - **Class T2** – T1 + Chip bypass (1-bit) for IR/DR Scans + hot-connection with no errors..
  - **Class T3** – T2 + Star operation (TAPs are paralleled) with series equivalent scans for test.
- **Advanced Operation** – adds capabilities addressing test and debug needs in complex environments:
  - **Class T4** – T3 + Scan with two pins with the Advanced Protocol. A number of protocol optimizations maximize scan performance. TDI(C)/TDO(C) pins are optional and may have programmable function when implemented.

### 49.3.3 1149.7 TAP signals

The TAP signals for the T0–T4 TAP.7 classes are listed in the following table. Signal names that end in “C” are associated with expanded functionality (beyond the functionality provided by the IEEE 1149.1 standard).

The table uses the following notation:

- M = Mandatory signal
- MC = Mandatory signal with expanded functionality
- O = Optional signal
- OC = Optional signal with expanded functionality

The CJTAG module provides T4 capability and implements the following pins:

- TCKC
- TDIC
- TDOC
- TMSC
- nTRST

**Table 49-2. TAP.7 signal list**

| Signal name                      | Description      | Class |    |    |    |    |
|----------------------------------|------------------|-------|----|----|----|----|
|                                  |                  | T4    | T3 | T2 | T1 | T0 |
| TCK/TCKC                         | Test clock       | MC    | M  | M  | M  | M  |
| TMS/TMSC                         | Test mode select | MC    | M  | M  | M  | M  |
| TDI/TDIC                         | Test data input  | OC    | MC | M  | M  | M  |
| TDO/TDOC                         | Test data output | OC    | MC | M  | M  | M  |
| nTRST                            | Test reset       | O     | O  | O  | O  | O  |
| Mandatory signal count (minimum) |                  | 2     | 4  | 4  | 4  | 4  |

### 49.3.4 TAP.7 architecture

The TAP.7 hardware architecture is shown in the following figure. It is described with the hardware layers listed below:

- **STL – System Test Logic** – Logic with 1149.1 compliant behavior and underlying TAP hierarchy (T0 capabilities). Provides an IEEE 1149.1 interface for the T0 TAP. 7. This logic is outside the CJTAG module.
- **RSU – Reset and Selection Unit** – A hardware layer that is placed between the APU, EPU, or STL and the TAP.7 signals (Added as an option to support the use of the Control Protocol) Provides reset and TAP.7 Controller selection services.
- **EPU – Extended Protocol Unit** – A hardware layer that is placed between the STL and the TAP.7 signals. (Added for T1, T2, and T3 capabilities). Provides an IEEE 1149.1 interface for the T1–T3 TAP.7s.
- **APU – Advanced Protocol Unit** – A hardware layer that is placed between the STL/EPU and the TAP.7 signals (added for T4 capabilities). Provides a T4 TAP.7 interface that is either narrow or wide, with the wide version providing an IEEE 1149.1 interface.

The CJTAG module supports T4 functionality that includes STL, RSU, EPU, and APU hardware layers.

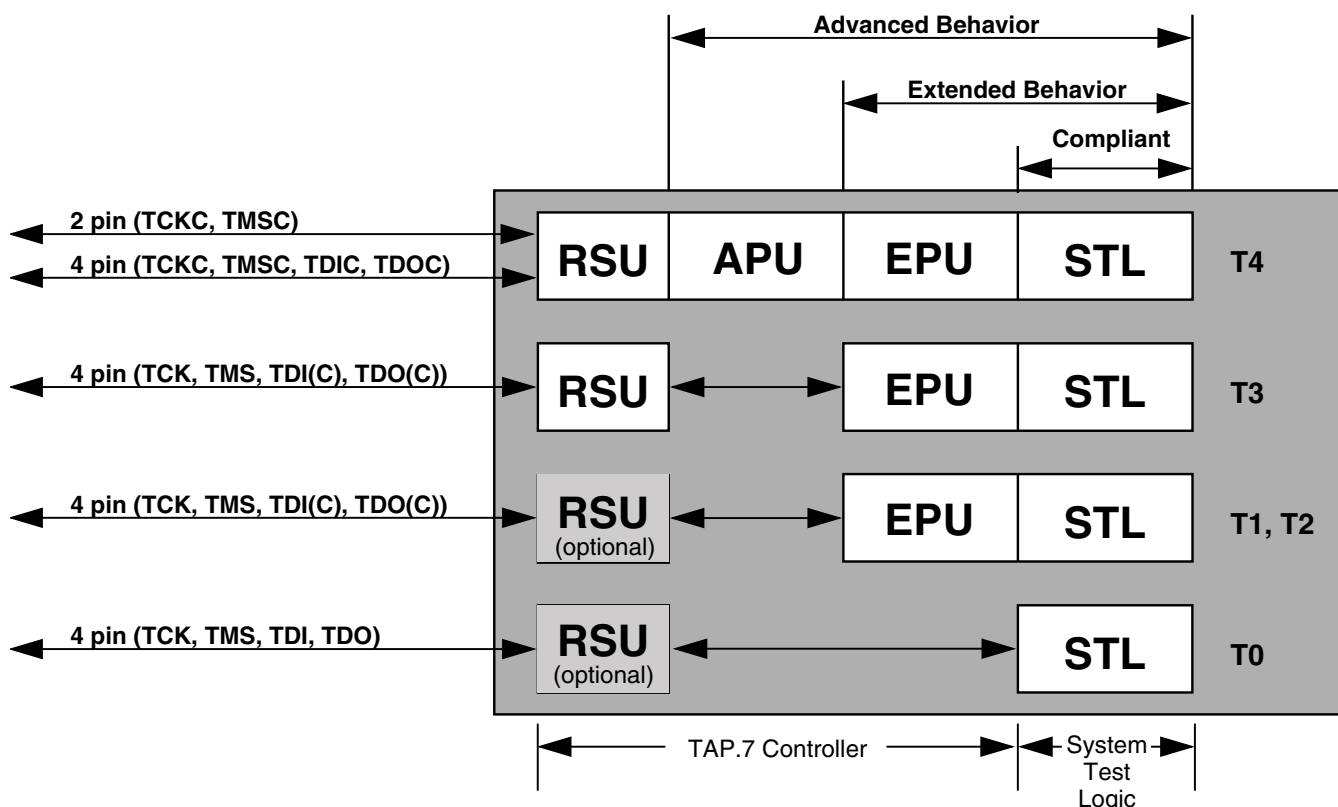


Figure 49-1. TAP.7 Controller Architecture

### 49.3.5 Protocols

The TAP.7 architecture utilizes the three protocols shown in the following figure.

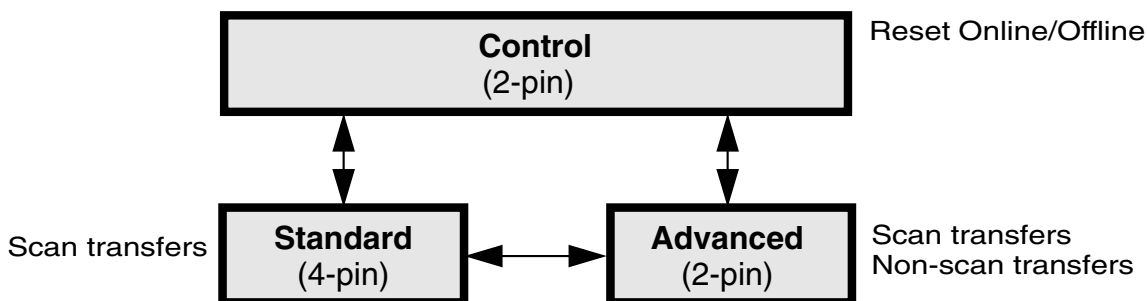


Figure 49-2. TAP.7 Behaviors

The Standard Protocol transfers data with the TMS(C) signal value being sampled with a TCK(C) edge. The Advanced and Control Protocols use both data transferred with the TMS(C) signal value being sampled with a TCK(C) signal edge and escape sequences. The Standard Protocol may be used to switch to the use of the Advanced Protocol and



vice versa. The Control Protocol can also be used to switch between the use of the Standard and the Advanced Protocol. The mandatory and optional deployment of these protocols with the TAP.7 classes is shown in the following table.

**Table 49-3. Class/protocol relationship**

| Class          | Protocol  |           |           |
|----------------|-----------|-----------|-----------|
|                | Standard  | Control   | Advanced  |
| <b>T0 – T2</b> | Mandatory | Optional  | N/A       |
| <b>T3</b>      | Mandatory | Mandatory | N/A       |
| <b>T4</b>      | Mandatory | Mandatory | Mandatory |

The CJTAG module implements T4 functionality and supports the Standard, Advanced, and Control Protocols.

#### 49.3.5.1 Standard Protocol

The Standard Protocol is the signaling defined by the IEEE 1149.1 standard. This protocol is also used to implement TAP.7 commands using only the TCK(C) and TMS(C) signals in a manner that does not require either TAP.7 controller instruction or data registers. The TAP.7 controller does not add bits to the scan paths of underlying technology. The Standard Protocol can be used to manage the TAP.7 controller functionality independent of the function associated with the availability of the TDI and TDO signals. The Standard Protocol supports STL data transfers only when these signals are present and provide the 1149.1 functionality defined by the IEEE 1149.1 standard.

#### 49.3.5.2 Advanced Protocol

The Advanced Protocol supports both STL data transfers and management of the TAP.7 controller functionality with both the 2-signal (narrow) and 4-signal (wide) configurations. The Advanced Protocol serializes the TMS, TDI, and TDO information used with the Standard Protocol using only the TCK(C) and TMS(C) signals. A number of additional scan formats provide bit sequences optimized for specific use cases. Each of these additional scan formats affects the TMSC signaling sequences.

### 49.3.5.3 Control Protocol

The Control Protocol uses Escape sequences to reset the TAP.7 controller, place a TAP.7 controller offline or online, and indicate End of Transmission (EOT) with the Advanced Protocol. The characteristics of this protocol provide Hot-Connect-Protection by requiring a preamble of alternating ones and zeroes preceding a selection escape when a TAP controller is started up offline.

## 49.4 Operating models

The following figure illustrates a simplified view of the CJTAG operating model.

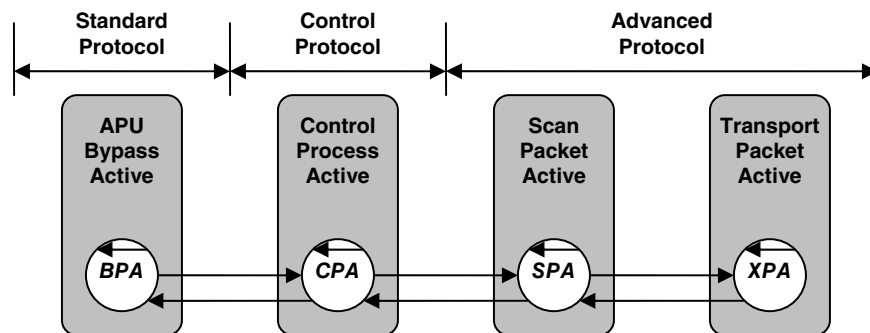


Figure 49-3. CJTAG operating model

## 49.5 CJTAG implementation summary

The CJTAG module is a T4 TAP.7 controller that implements the following functions.

### 49.5.1 T0 functions

- IDCODE. - implemented in chip level TAPC (JTAGC module)
- 1149.1 Compliance at start-up
- Isolating / Excluding of embedded TAPs – implemented in chip level TAPC (JTAGC module)
- Operation with other technologies and scan topologies
- Multiple embedded TAPs

### 49.5.2 T1 functions

- EPU Command processing
- EPU Class-Specific Registers
- Asserting a test reset to the STL (TRESET Register)

### 49.5.3 T2 functions

- “Super Bypass” function – JScan0, JScan1, JScan2 scan formats. Allows the STL to be coupled or decoupled.

### 49.5.4 T3 functions

- Generation of a TAP.7 controller reset with a Reset Escape Sequence
- JScan3 Scan Format

### 49.5.5 T4 functions

- TMSC sampling on rising edge or falling edge of TCKC
- MScan (For test and debug)
- OScan 0-7
- TDIC and TDOC / Alternate pin function multiplexing (wide 4 only)

## 49.6 Ancillary services

### 49.6.1 Overview

Ancillary services are pertinent to any TAP.7 controller implemented with an RSU, such as the CJTAG module.

Ancillary services include:

- Resets
- Start-up options

- Escape sequences
- TAPC state machine

## 49.6.2 Resets

The TAP.7 controller reset function expands the reset function provided by the IEEE 1149.1 standard to provide additional reset types. It supports six reset types, Type-0 – Type-5, listed below. It can be implemented with as few as two reset types (Type-2 and Type-4), or as many as six reset types (all types) of reset inputs: This is determined by the class and options implemented.

The CJTAG module utilizes Type-0, Type-2, Type-3, Type-4, and Type-5 resets.

- **Type-0** – Reset of the TAP.7 by power management logic (power-on reset)
- **Type-1** – Chip level generated Test Reset. (equivalent to Type-2)
- **Type-2** – Externally generated Test Reset (nTRST/nTRST\_PD pin)
- **Type-3** – TAP.7 controller generated reset
- **Type-4** – *Test-Logic-Reset* state
- **Type-5** – TAP.7 Controller TReset Register reset of the CLTAPC.

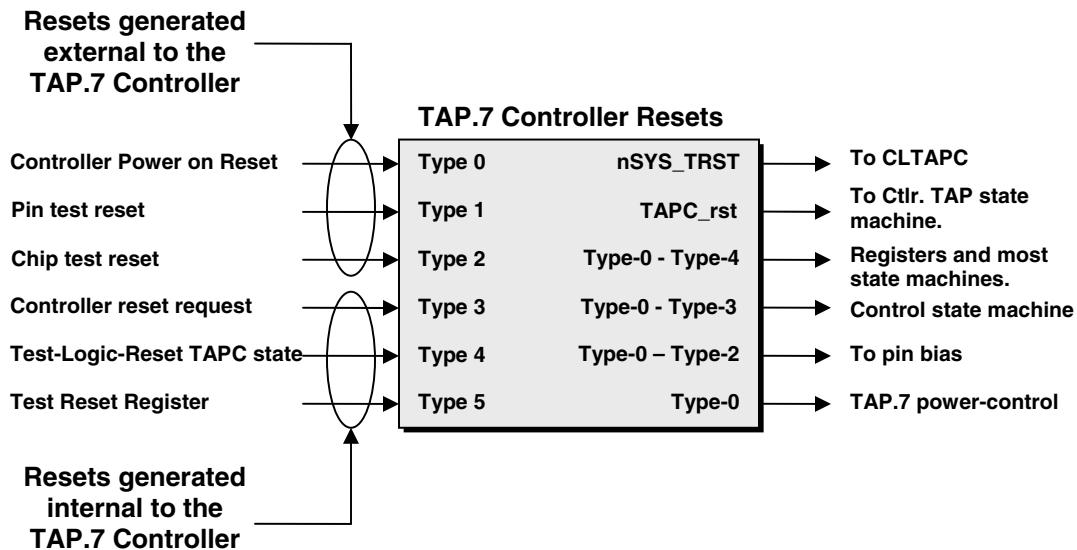


Figure 49-4. Conceptual view of the EPU reset logic and state machines

### 49.6.2.1 Type 5 reset

A Type-5 reset affects only the STL via the nSYS\_TRST pin. It is asserted when the TRESET Register value is a logic 1. This register is set to its inactive state by reset Types 0-4.

### 49.6.2.2 Type 4 reset

A Type-4 reset is created by the *Test-Logic-Reset* state. It does not affect the TAP.7 TAPC state machine state as this state is the source of this reset, nor does it cause the assertion of the nSYS\_TRST signal. It does however affect the coupling and decoupling of the STL. The Type-4 reset initializes the TAP.7 registers and most controller state machines. It affects but does not initialize the control state machine managing TAP.7 controller selection. It also establishes the default operating conditions.

### 49.6.2.3 Type 3 reset

A Type-3 reset is created by reset requests generated by the TAP.7 controller. A Type-3 reset performs the functions associated with a Type-4 reset and creates the *Test-Logic-Reset* state.

A Type-3 reset is generated in response to reset requests generated by EPU and APU logic

- A reset escape sequence
- Reset protocol sequences (delay packet directive, check packet directive)

A Type-3 reset request is generated with a reset escape sequence. This function is similar to but not the same as the reset function generated by a test reset pin. The Advanced Protocol includes certain bit patterns that initiate Type-3 resets. These bit patterns assist in the initialization of the TAP.7 controller when the TCKC signal is sourced by the TS and the external tool/TS connection is broken. The external tool may also generate these sequences, if desired, while the external tool/TS connection remains intact. These reset requests cause a Type-3 reset that is one clock wide.

With this being the case, consideration may be given to integrating the CJTAG module without the use of the Test Reset signal (nTRST/nTRST\_PD pin).

#### 49.6.2.4 Type 2 reset

A Type-2 reset is created by chip level logic to initialize the TAP.7 controller when TAP.7 power control is not implemented. It is unimplemented otherwise. It performs all of the functions of the Type-3 reset in addition to initializing the escape sequence detection logic.

#### 49.6.2.5 Type 1 reset

A Type-1 reset is created with either the nTRST or nTRST\_PD pin. It is not created when neither of these pins is implemented. It performs all of the functions of the Type-2 reset. It initiates TAP.7 controller power down provided the TAP.7 power control mode permits power down.

#### 49.6.2.6 Type 0 reset

A Type-0 reset is implemented when TAP.7 controller power down is implemented. It performs all of the functions of a Type-1 reset and also initializes the TAP.7 power control logic.

#### 49.6.2.7 Reset State Machine

The Reset State Machine assists in the translation of the reset inputs to the reset outputs. It handles Type-3 reset controller reset requests. The state machine state assures Type-0, Type-1, and Type-2 resets are forwarded to the STL without modifying their duration.

This state machine allows a Type-3 reset only when its state is “T3 Reset” (two TCK(C) falling edges have occurred in the absence of a Type-0, Type-1, and Type-2 reset).

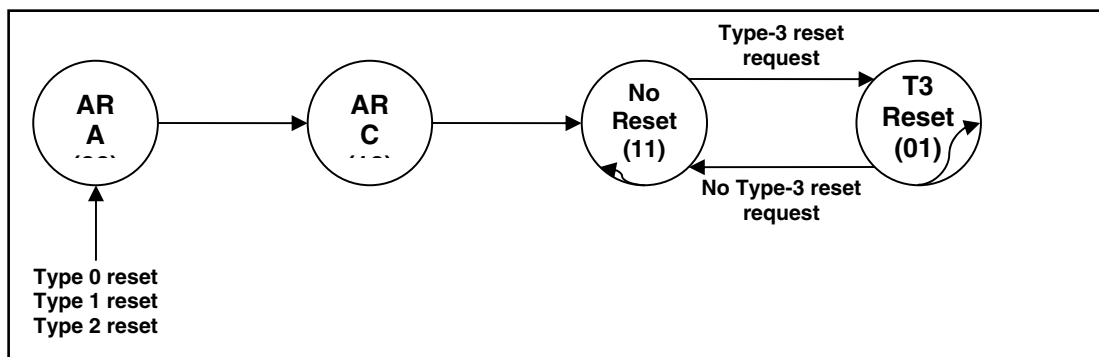


Figure 49-5. Reset state machine

**Table 49-4. Reset state descriptions**

| Reset state |                        | Description                                        |
|-------------|------------------------|----------------------------------------------------|
| ARA         | Async. Reset Asserted  | An async. reset has occurred or is occurring       |
| ARC         | Async. Reset Completed | An async. reset has occurred and has been released |
| No Reset    | No Reset               | No Reset                                           |
| T3 Reset    | Type-3 Reset           | A Type-3 reset request has been processed.         |

Type-0, Type-1, and Type-2 resets are fully asynchronous. These resets are passed directly to nSYS\_TRST without modification. The Type-3, Type-4, and Type-5 resets are fully synchronous.

The effects of the TAP.7 controller reset types are shown in the following table.

**Table 49-5. Reset effects**

| Reset  | CLTAPC | Pin-Hi-Z | State Machines other than CSM | Control State Machine (CSM) | TAPC SM State | Escape Sequence Detection | Power Control State |
|--------|--------|----------|-------------------------------|-----------------------------|---------------|---------------------------|---------------------|
| Type-5 | Yes    | –        | –                             | –                           | –             | –                         | –                   |
| Type-4 | –      | Yes      | Yes                           | –                           | –             | –                         | –                   |
| Type-3 | Yes    | Yes      | Yes                           | Yes                         | Yes           | –                         | –                   |
| Type-2 | Yes    | Yes      | Yes                           | Yes                         | Yes           | Yes                       | –                   |
| Type-1 | Yes    | Yes      | Yes                           | Yes                         | Yes           | Yes                       | –                   |
| Type-0 | Yes    | Yes      | Yes                           | Yes                         | Yes           | Yes                       | Yes                 |

## 49.6.3 Start-up Options

### 49.6.3.1 Overview

The TAP.7 start-up options are shown in the following table. The only startup option currently supported by the CJTAG module is 1149.1 Compliant.

**Table 49-6. Start-up Options**

| Start-up Option          | Description                                         |
|--------------------------|-----------------------------------------------------|
| <b>1149.1 Compliant</b>  | TAP.1 signal functionality, the CLTAPC is coupled   |
| <b>1149.1 Compatible</b> | TAP.7 signal functionality, the CLTAPC is decoupled |

*Table continues on the next page...*

**Table 49-6. Start-up Options (continued)**

| Start-up Option                      | Description                                                                                                                                                                               |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>1149.1 Protocol Compatible</b>    | No, or alternate TDIC/TDOC signal functionality, the CLTAPC is decoupled.                                                                                                                 |
| <b>Offline-at-Start-up (Dormant)</b> | The TAP.7 controller is offline and awaits synchronization to external tool operation so as to be placed online, the CLTAPC is decoupled after its state is moved to <i>Run-Test/Idle</i> |

### 49.6.3.2 1149.1 Compliant start-up

Start-up behavior with the 1149.1 Compliant behavior start-up option:

- 1149.1-specific behavior / Standard protocol
- The TAPC finite state machine operated in lock step with the CLTAPC finite state machine.
- The TDIC and TDOC pins are present and provide the TDI and TDO function.
- The CLTAPC is coupled at start-up.

With this start-up option, the TAP.7 controller may be programmed and the scan topology may be interrogated. If the scan topology is known to be the Series Scan Topology at power-up, the TAP.7 can be operated as though it is a TAP.1.

**Table 49-7. Reset values for TAP.7 startup options**

| Start-up Option  | Type-0 – Type-3 Reset? | Default TAP.7 and TAP.7 controller characteristics |                |               |        |                       |
|------------------|------------------------|----------------------------------------------------|----------------|---------------|--------|-----------------------|
|                  |                        | TAP.7 Controller online                            | CLTAPC Coupled | SGC Reg. == 1 | SCNFMT | Initial TMSC pin bias |
| 1149.1 Compliant | x                      | Yes                                                | Yes            | Yes           | JScan0 | PU                    |

### 49.6.4 RSU operation

The RSU uses TCKC and TMSC with the control protocol for configuration control.



### 49.6.4.1 General Operation

From a very high level, the RSU enables the remainder of the TAP.7 functionality when the TAP.7 controller is Online and disables the remainder of the TAP.7 functionality otherwise. The RSU either consumes or discards the TMS(C) information while the TAP.7 controller is Offline. The relationship of the escape sequence detection, technology selection logic, and the underlying technology is shown in the following figure.

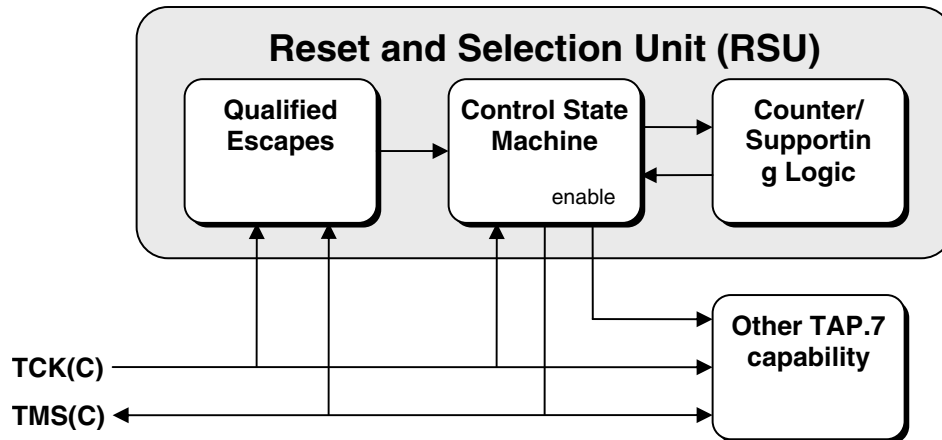


Figure 49-6. Conceptual view of technology selection mechanism

### 49.6.4.2 Common Signaling Across Technologies

With the TAP.7 architecture, the signaling conventions chosen for technology selection is multiple data transitions of the TMS(C) signal while the TCK(C) signal is a logic 1. The common signaling space is available when the external tool sources the TCK(C) signal as it is capable of creating these signaling conventions. It is unavailable when the TS sources the TCK(C) signal as the external tool cannot stop the TCKC signal at a logic 1 to create these signaling conventions.

### 49.6.4.3 Escape Sequence Detection

TAP.7 TAPC interprets the count of the number of TMS(C) edges while TCK(C) is a logic 1 as one of four escape sequences. Each escape sequence has a different function and TMSC edge count. These escape sequences, their edge counts, and their functions are described below:

- **Custom** (2 or 3 edges) – Ends scan and transport transfers with a TAP.7 controller, may be used for other purposes with another technology.
- **Reset** (> 7 edges) – Resets all technologies (generates a Type-3 TAP.7 TAPC reset)

The external tool creates an escape sequence by generating one or more TMS(C) edge pairs while the TCK(C) signal is a logic 1 value. Even and odd TMS(C) edge counts beginning with two are given the same meaning.

An escape sequence:

- Can be detected only when a Type-0, Type-1 or Type-2 reset is **not** asserted.
- Begins and ends while the TCK(C) is a logic 1.
- Uses TMS(C) as a clock while the TCK(C) is a logic 1.
- Overlays additional control information onto the normal information that is transferred with the TCK(C) and the TMS(C) without changing the normal information.
- Must be detected when TCK(C) is a logic 1 regardless of the TCK(C) and TMS(C) drive histories.

#### 49.6.4.3.1 Custom escape sequence

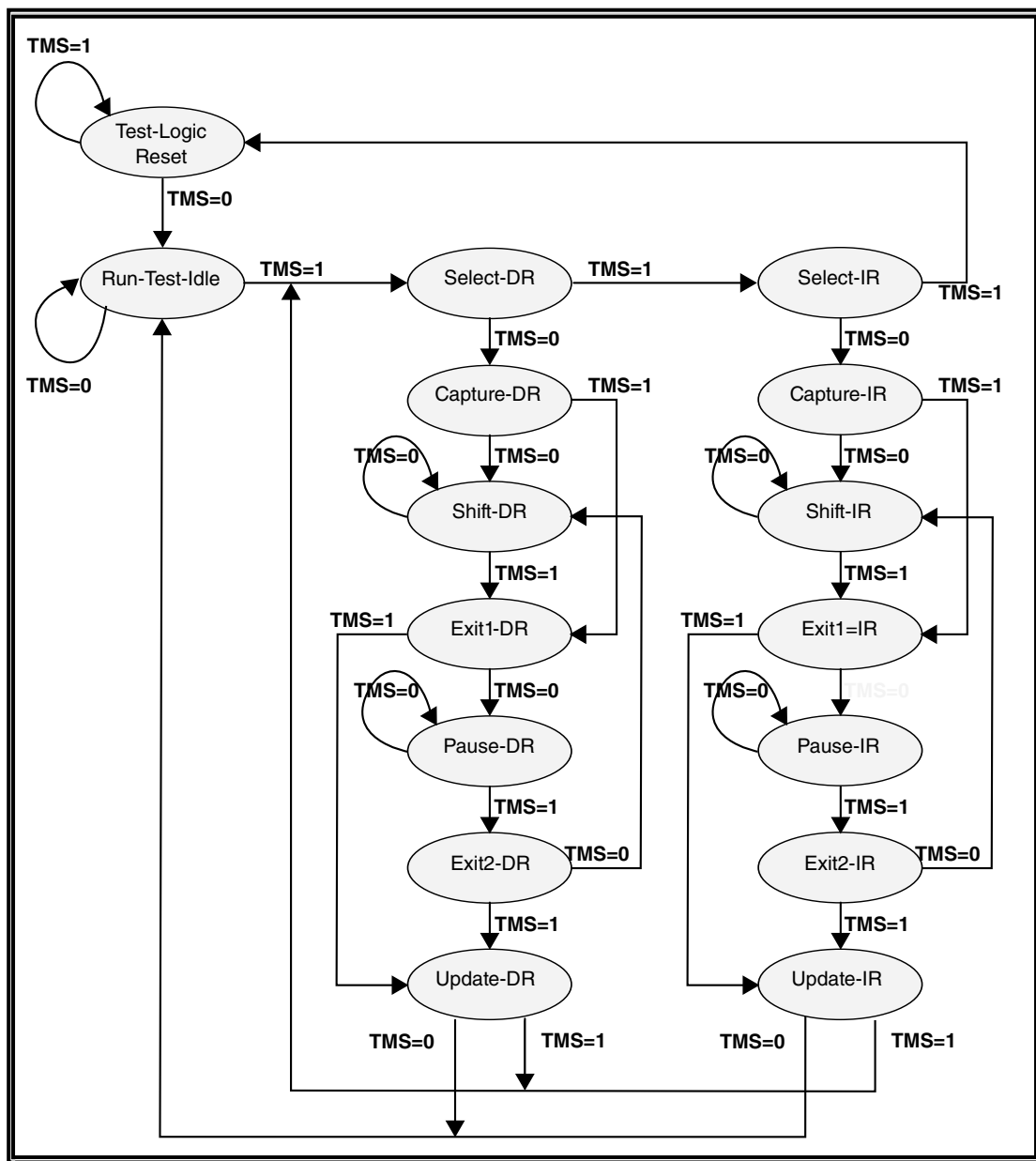
An online technology can use the custom escape sequence in any manner it chooses. It is used as the End of Transmission (EOT) of certain data types (formats OScan4-7) with T4 and above TAP.7s.

#### 49.6.4.3.2 Reset escape sequences

A reset event initializes all technologies sharing the TCK(C) and TMS(C) connectivity. A reset escape sequence resets the technology. It is recommended that this reset occur with the falling edge of TCKC. With a TAP.7 controller a reset escape sequence generates a Type-3 reset beginning with the falling edge of TCK(C) following the asynchronous detection of the event.

### 49.6.5 TAPC State Machine

T1 and above TAP.7s require knowledge of the TAPC state machine state. With T2 and above TAPs this knowledge must be developed independently of the CLTAPC. This is generally accomplished by implementing a TAPC state machine within the EPU. This TAPC state machine provides the function of the state machine within a standard 1149.1 TAPC. It is recommended that this state machine is implemented to change state on the falling edge of TCK(C) as this has numerous advantages to a TAP.7 controller. The EPU does not contain 1149.1 Instruction or Data Registers. The APU controls the state changes of this TAPC state machine with T4 and above TAP.7s.



### Figure 49-7. Falling edge TAP controller state machine

## 49.7 EPU (Extended Protocol Unit) Operation

### 49.7.1 EPU Operation

The EPU adds functionality to the BYPASS and IDCODE instructions.

### 49.7.1.1 Zero-bit DR-Scans (ZBS)

A zero-bit DR-Scan (ZBS) is a TAPC-state sequence that begins with the *Select-DR-Scan* state and ends with an exit from the *Update-DR* state without an intervening *Shift-DR* state. The external tool controls the EPU using ZBSs.

The use of ZBSs can begin immediately following the *Test-Logic-Reset* state since this state initializes the Instruction Registers of TAPs with either the *BYPASS* or *IDCODE* instructions.

### 49.7.1.2 Utilizing ZBSs for TAP.7 Controller Functionality

Beginning with a ZBS count of zero, the ZBS count is incremented with each consecutive occurrence of a ZBS without encountering a *Shift-DR* state. This count cannot be incremented past seven. *Run-Test/Idle* is the only state that may occur between consecutive occurrences of ZBSs without terminating the count of consecutive ZBSs.

### 49.7.1.3 Locking the ZBS Count

When a DR-Scan containing a *Shift-DR* state occurs, and the ZBS count is greater than zero, the ZBS count is not incremented when the *Update-DR* state is reached. It is instead locked at its current value. ZBSs occurring after locking the ZBS count do not affect the locked ZBS count. Locking the ZBS count is the equivalent of storing the count for subsequent use.

### 49.7.1.4 Control levels

Locking the ZBS count activates a control level that is equal to the locked ZBS count (1-7) when ZBSs are being used for TAP.7 functionality and merely locks the ZBS count otherwise.

1. ZBSs may be used by the STL or another function
2. TAP.7 command generation
3. Reserved control level
4. Accesses to optional TAP.7 scan paths
5. Accesses to optional TAP.7 scan paths

6. Force offline operation (optional)
7. External tool uses

Control level one may be used by the STL since there is no TAP.7 controller functionality associated with this control level.

Control level two enables DR-Scans to create TAP.7 commands. A portfolio of these commands supports the deployment of TAP.7 controllers in the Series or Star Scan Topologies. These commands manage TAP.7 controller registers and perform other functions.

Control level three is reserved for use with a subsequent specification revision.

Control levels four and five provide access to custom (private) scan paths when access to these paths is permitted by a TAP.7 register bit used for this purpose. The EPU provides a 1-bit scan path otherwise. No other EPU functionality is associated with these control levels. The CJTAG module does not support the use of this control level.

Control level six may be optionally used to force offline operation with a TAP.7 controller that supports this capability. The EPU provides a 1-bit scan path for this control level. The CJTAG module does not support the use of this control level.

Control level seven is reserved for external tool use. With this control level, external tool capabilities may be managed using the same infrastructure used to manage TAP.7 controller capability. The EPU provides a 1-bit scan path for this control levels.

#### 49.7.1.5 Exiting a Control Level

A control level is exited when one of the following occurs:

- The *Select-IR-Scan* TAP.7 controller state
- The *Test-Logic-Reset* state
- Certain TAP.7 controller commands (Exit Control Level – ECL register bit)
- An event that synchronizes the operation of TAP.7 controllers

#### 49.7.1.6 Zeroing the ZBS Count

Events causing an exit from a control level also zero the ZBS count. Scan Selection Directives (SSDs) associated with T3 and above TAP.7s zero the ZBS count when the control level has not been locked.

### 49.7.1.7 Utilizing ZBSs within the STL

After the *Test-Logic-Reset* state, ZBSs are used for basic EPU control. ZBSs that are intended for TAP.7 use are henceforth called "**TAP.7 ZBSs**". ZBSs that are intended for other uses are henceforth called "**Other ZBSs**". Two TAP.7 registers enable the STL's use of ZBSs. Storing these registers causes a control level exit, zeroes the ZBS count, and enables the use of Other ZBSs. Storing the Suspend Register (**SUSPEND**) suspends the use of control levels. Storing the ZBS Inhibit Register (**ZBSINH**) inhibits ZBS detection.

When either of these methods is used, the *Test-Logic-Reset* state enables the use of TAP.7 ZBSs in addition to initializing the TAP.7 controller.

The operation of the TAP.7 controller changes with the use of these two paths.

When the System Path is used:

- A count of eight consecutive ZBSs is used to create a logic 0 SUSPEND Register value.
- The CLTAPC supplies the TAP.7 scan path and controls the drive of TDO(C), provided it is not bypasses.
- The EPU supplies the TAP.7 scan path (a one bit bypass) when the CLTAPC is bypassed.
- Functions supporting Boundary Scan in a Star Scan Topology are enabled.

When the EPU Path is used:

- When the control level has not been locked, the TDO(C) signal remains high-impedance.
- The command control level may be created
- When commands are used, the TDO(C) signal remains high-impedance except during *Shift-DR* states within CR-Scans, and remains high impedance at other times.
- Auxiliary scan paths may be accessed with control levels four and five.

## 49.7.2 EPU Registers

### 49.7.2.1 Global

A global register is a register whose value is stored at the same time and with the same value as registers with the same name in all TAP.7 TAPCs sharing an external tool connection. They are stored using TAP.7 TAPC commands dedicated for this purpose. These commands store a single global register. Global registers manage TAP.7 TAPC functions that affect the synchronized operation of a TAP.7 TAPC. An example of a global register is the Scan Format (SCN\_FMT) Register used to define the TAPC signal protocol.

A second means of storing all global registers is available. The selection of the TAP.7 TAPC with the technology selection function stores either the Scan Format register value (short form) or all global registers (long form). Storing all global registers provides a simple means to synchronize the values of all global registers when Technology Selection causes a non-operating TAP.7 TAPC to become operational while other TAPCs are operational.

### 49.7.2.2 Local

Local registers manage TAP.7 TAPC functions that do not affect the synchronized operation of TAP.7 TAPCs. Local registers are conditionally stored using TAP.7 controller commands dedicated for this purpose. These commands provide for changing the value of these registers one at a time with a unique value in a Series Scan Topology based on the TAPC's position on the scan chain. Local registers are stored with a unique value in a Star Scan Topology based on a TAPC address assigned to the TAPC.

Registers programmed with commands are described in this section. The CJTAG module implements all mandatory T4 registers as well as the optional APFC, TRESET, and RDBACK0 registers. There are 16 mandatory and 10 optional registers that can be grouped in classes as shown in Read-only – Configuration and register read back..

- **Control** – TAP.7 controller behaviors and characteristics.
- **Options** – Control of optional functions.
- **Select** – Controls the selection of a TAP.7 for Scan and execution of conditional commands.
- **Read-only** – Configuration and register read back.

**Table 49-8. TAP.7 controller register list (managed with commands)**

| Register type | Global/local | Width | Register mnemonic | Name                     |
|---------------|--------------|-------|-------------------|--------------------------|
| Control       | Global       | 1     | ECL               | Exit Control Level       |
|               |              | 1     | SUSPEND           | Suspend                  |
|               |              | 1     | ZBSINH            | ZBS Detect Inhibit       |
|               |              | 5     | SCNFMT            | Scan Format              |
|               |              | 1     | SSDE              | SSD Enable               |
|               |              | 2     | DLYC              | Delay control            |
|               |              | 2     | RDYC              | Ready Control            |
| Options       | Local        | 1     | TRESET            | Test Reset               |
|               |              | 2     | APFC              | Aux. Pin Func. Cntl.     |
| Select        | Local        | 1     | CGM               | Cond. Group Member       |
|               |              | 1     | SGC               | Scan Group Candidate     |
|               |              | 1     | SEDGE             | Sampling Edge            |
| Read-only     | Local        | 32    | RDBACK0           | Read-back 0              |
|               |              | 32    | RDBACK1           | Read-back 1              |
|               |              | 32    | CNFG0             | Configuration Register 0 |

M = Mandatory register

O = Optional register

### 49.7.2.3 Register Descriptions

#### 49.7.2.3.1 Exit Control Level (ECL)

Storing this register zeroes the ZBS count and unlocks the control level.

This register is set using the STMC command. It remains logic 1 only momentarily before immediately returning to logic 0.

This register is cleared by a global register load.

#### 49.7.2.3.2 Suspend (SUSPEND)

Setting this register suspends the use of control levels. When this register is a logic 1, the "System Path" (STL) is used.

**1** – Suspend use of control levels. Enables "System Path."

**0** – Enable use of control levels.



SUSPEND is set with the STMC command.

This register is cleared when 8 consecutive ZBSs are detected without the locking of the ZBS count.

This register is cleared by a global register load.

#### 49.7.2.3.3 ZBS Inhibit (ZBSINH)

Storing to ZBSINH inhibits ZBS detection.

**1** – Inhibit ZBS detection.

**0** – Enable ZBS detection.

ZBSINH is set with the STMC command.

This register is cleared by a global register load.

#### 49.7.2.3.4 Scan Format (SCNFMT[4:0])

Set using Store Scan Format (STFMT) command.

Read using Scan String (SCNS) command.

**Table 49-9. Scan Format Register Values**

| TAPC.7 Class | SCNFMT | Description                                                                                                                                                                                                                                  |
|--------------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| T2–T4        | 00000  | <b>JScan0:</b> <ul style="list-style-type: none"> <li>• 1149.1 Compliant Operation.</li> <li>• Requests the coupling of the CLTAPC when the Run-Test/Idle state is reached.</li> </ul>                                                       |
|              | 00001  | <b>JScan1:</b> <ul style="list-style-type: none"> <li>• 1149.1 Compatible Operation</li> <li>• Requests the decoupling of the CLTAPC when the Run-Test/Idle state is reached.</li> </ul>                                                     |
|              | 00010  | <b>JScan2:</b> <ul style="list-style-type: none"> <li>• 1149.1 Compatible Operation.</li> <li>• Requests the coupling or decoupling of the CLTAPC when the Run-Test/Idle state is reached based on the value of the CACT register</li> </ul> |
| T3–T4        | 00011  | <b>JScan3:</b>                                                                                                                                                                                                                               |

*Table continues on the next page...*

Table 49-9. Scan Format Register Values (continued)

| TAPC.7 Class | SCNFMT | Description                                                                                                                                                                                                                                                                                                             |
|--------------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|              |        | <ul style="list-style-type: none"> <li>• Non-compatible 1149.1 Operation</li> <li>• Allows the detection of a T2 TAP.7 controller that is deployed in a Star-4 topology.</li> <li>• Enables TDOC signal behavior that prevents drive conflicts in a Star-4 Scan Topology.</li> <li>• Enables the use of SSDs</li> </ul> |
| T4           | 01000  | <b>OScan0</b> <ul style="list-style-type: none"> <li>• Data rate dependent components</li> <li>• Single TAP.7 communication</li> <li>• Debug and Test applications</li> </ul>                                                                                                                                           |
|              | 01001  | <b>OScan1</b> <ul style="list-style-type: none"> <li>• TAP.1 components</li> <li>• Single/Multi-TAP.7 communication</li> <li>• Debug and Test applications</li> </ul>                                                                                                                                                   |
|              | 01010  | <b>OScan2</b> <ul style="list-style-type: none"> <li>• TAP.1 components</li> <li>• No TDI or TDO in non-shift states</li> <li>• Debug applications</li> </ul>                                                                                                                                                           |
|              | 01011  | <b>OScan3</b> <ul style="list-style-type: none"> <li>• TAP.1 components</li> <li>• No TDI in non-shift states</li> <li>• No TDO in any state</li> </ul>                                                                                                                                                                 |
| T4           | 01100  | <b>OScan4</b> <ul style="list-style-type: none"> <li>• Data rate dependent components</li> <li>• Single TAP.7 communication</li> <li>• Debug and Test applications</li> </ul>                                                                                                                                           |
|              | 01101  | <b>OScan5</b> <ul style="list-style-type: none"> <li>• TAP.1 components</li> <li>• Single/Multi-TAP.7 communication</li> <li>• Debug and Test applications</li> </ul>                                                                                                                                                   |
|              | 01110  | <b>OScan6</b> <ul style="list-style-type: none"> <li>• TAP.1 components</li> <li>• No TDI or TDO in non-shift states</li> <li>• Debug applications</li> </ul>                                                                                                                                                           |

Table continues on the next page...

**Table 49-9. Scan Format Register Values (continued)**

| TAPC.7 Class | SCNFMT        | Description                                                                                                                                                                                                                                         |
|--------------|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|              | 01111         | <b>OScan7</b> <ul style="list-style-type: none"> <li>• TAP.1 components</li> <li>• No TDI in non-shift states</li> <li>• No TDO in any state</li> </ul>                                                                                             |
|              | 10000         | <b>MScan</b> <ul style="list-style-type: none"> <li>• Data rate dependent components</li> <li>• Multi TAP.7 communication</li> <li>• Directed CID assignment</li> <li>• Virtually any IP with compliant or non-compliant 1149.1 behavior</li> </ul> |
|              | 10001 – 11111 | <b>Reserved</b>                                                                                                                                                                                                                                     |

The use of a Scan Format value other than 00000b – 10000 shall not change the value of the SCNFMT Register.

#### 49.7.2.3.5 Scan Selection Directive Enable (SSDE)

This register specifies whether the Scan Selection State may be managed using the scan selection directives.

**0** – Scan selection directives are not enabled.

**1** – Scan selection directives are enabled.

This register is set using the STMC command.

This register is read using SCNS (Scan String), RDBACK0.

#### 49.7.2.3.6 Delay Control (DLYC[1:0])

The Delay Control register provides a means for the external tool to delay the completion of an SP. This delay is essentially an external tool stall. This delay may be zero, one, two, or n TCKC periods. This delay is especially useful when the TCKC is sourced by the TS and a Standard-to-Advanced Protocol adapter is added to an external tool supporting only the Standard Protocol. It allows an external tool design additional time to receive TDO, advance the external tool TAPC state, receive TMS and TDI information, and begin generation of a new SP payload. The DLYC register should be programmed before the use of the Advanced Protocol begins after the use of the Standard Protocol.

Set using the STMC command.

Read using SCNS, RDBACK0

Bits 18-17 of the "Global Register."

#### 49.7.2.3.7 Ready Control (RDYC[1:0])

The Ready Control register defines behavior exhibited in the SP payload output bit frames when the STL stall opportunities (RDY bits) are included as control information. This register defines the number of additional bits inserted in these bit frames. These bits provide more time for a high-performance external tool to ascertain whether the TAP.7 controller is ready to complete the SP payload. The use of these bits makes a high-performance external tool design easier as input and output buffer delays play less of a role in limiting the TAP.7 controller performance. The RDYC register should be programmed before the use of the Advanced Protocol begins.

Set using the STMC command.

Read using SCNS, RDBACK0.

Bits 18-17 of the Global register state.

#### 49.7.2.3.8 Test Reset (TRESET)

Asserts a test reset to the STL.

**0** – The nSYS\_TRST and SYS\_TMS signals presented to the STL is not influenced by this bit.

**1** – The nSYS\_TRST presented to the STL is asserted and the SYS\_TMS signal is a logic 1.

Set using the STC1 command.

The effects of the TRESET Register may be modified by a private TAP.7 controller register. When this private register is unimplemented or has a value of zero, the TRESET register causes the maximum initialization of the STL with nSYS\_TRST and the SYS\_TMS logic 1 value. Otherwise, the value of the private register may modify the function of the TRESET Register bit.

When the TRESET Register is a logic 1, the STL scan path appears broken as the state of CLTAPC remains *Test-Logic-Reset* while the EPUTAPC state progression continues. When the operation of the EPUTAPC and CLTAPC controllers is coupled, as with a T1 TAP.7, the state of these two TAPCs may be resynchronized to the *Run-Test/Idle* state by:

- An STC1 Command that sets the TRESET Register value to a logic 0.
- A stay in the Run-Test/Idle state of two or more SYS\_TCK periods immediately following the Update-DR state of this command

A failure to follow the guidelines is considered a programming error and will result in erroneous system operation. The EPUTAPC and CLTAPC states will not be synchronized as they progress through the state diagram.

#### 49.7.2.3.9 Auxiliary Pin Function Control (APFC[1:0])

This register enables TDI and TDO alternate functions when 2-pin TAP.7 operation is selected.

**0x** – TDI and TDO have JTAG function.

**1x** – TDI and TDO have alternate function.

Set using STC2 command.

Read using SCNS, RDBACK0.

#### 49.7.2.3.10 Conditional Group Member (CGM)

Used by STC1, STC2, STTESTM, and EXC3 – EXC0 commands as the “condition” for updating registers.

Set and cleared using MCM and SCNB commands.

Read using SCNS, RDBACK0.

#### 49.7.2.3.11 Scan Group Candidate (SGC)

This register determines whether the STL is coupled when the *Run-Test/Idle* state is reached when scan formats other than JScan0 and JScan1 are used.

- **0** – Decoupling action.
- **1** – Coupling action.

Set and cleared using STMC, MSC, and SCNB commands.

Read using SCNS, RDBACK0.

#### 49.7.2.3.12 Sampling Edge (SEEDGE)

The Sampling Edge register defines the TCKC edge used to sample the TMSC input.

## Register Descriptions

- **0** – Sample TMSC on the TCKC falling edge.
- **1** – Sample TMSC on the TCKC rising edge.

Set and cleared using STC1 command.

Read using SCNS, RDBACK0

### 49.7.2.3.13 Read Back 0 (RDBACK0)

Non-transport related register information.

Read using SCNS command.

Unimplemented registers shall read back as "0".

**Table 49-10. RDBACK register format**

| Bit   | Width | Register Mnemonic | Name                             |
|-------|-------|-------------------|----------------------------------|
| 4:0   | 5     | SCNFMT            | Scan Format                      |
| 6:5   | 2     | PWRMODE           | Power-control Modes              |
| 7     | 1     | FRESET            | Functional Reset                 |
| 8     | 1     | TRESET            | Test Reset                       |
| 9     | 1     | SGC               | Scan Group Candidate             |
|       |       | CGM               | Conditional Group Member         |
| 11    | 1     | SSDE              | Scan Selection Directive Enable  |
| 13:12 | 2     | TOPOL             | Topology                         |
| 14    | 1     | SREDGE            | Sampling Edge                    |
| 16:15 | 2     | DLYC              | Delay Control                    |
| 18:17 | 2     | RDYC              | Ready Control                    |
| 20:19 | 2     | APFC              | Auxiliary Pin Functional Control |
| 22:21 | 2     | STCKDC            | SYS_TCK Duty Cycle               |
| 31:23 | 11    | Reserved          | Read as a logic 0                |

### 49.7.2.3.14 Configuration Register 0 (CNFG0)

**Table 49-11. Configuration Register 0 Format**

| Field           | Bit # | Width | Name       | Description                    |
|-----------------|-------|-------|------------|--------------------------------|
| CNFG<br>(Class) | 3:0   | 4     | CLASS[3:0] | TAP.7 Class                    |
|                 |       |       |            | 0000 – TAP.1 or TAP.7 Class T0 |
|                 |       |       |            | 0001 – Class T1                |
|                 |       |       |            | 0010 – Class T2                |
|                 |       |       |            | 0011 – Class T3                |
|                 |       |       |            | 0100 – Class T4                |

*Table continues on the next page...*

**Table 49-11. Configuration Register 0 Format (continued)**

| Field              | Bit # | Width | Name              | Description                               |
|--------------------|-------|-------|-------------------|-------------------------------------------|
|                    |       |       |                   | 0101 – Class T5                           |
| CNFG<br>(Revision) | 7:4   | 4     | REV[3:0]          | TAP.7<br>specification<br>revision.       |
|                    |       |       |                   | 0000 –<br>Reserved                        |
|                    |       |       |                   | 0001 – IEEE Std<br>1149.7-2008            |
|                    |       |       |                   | 0010 – 1111 -<br>Reserved                 |
| CNFG<br>(Format)   | 11:8  | 4     | CNFGFMT[3:0]      | Configuration<br>Register Format          |
|                    |       |       |                   | xxx1 –<br>CNFG0[31:12]<br>are implemented |
|                    |       |       |                   | xx1x – CNFG1<br>implemented               |
|                    |       |       |                   | x1xx – CNFG2<br>implemented               |
|                    |       |       |                   | 1xxx – CNFG3<br>implemented               |
| T1<br>Options      | 12    | 1     | RDBKS             | RDBACK<br>supported                       |
|                    |       |       |                   | 0 – Unsupported                           |
|                    |       |       |                   | 1 – Supported                             |
|                    | 13    | 1     | TRESETS           | TRESET<br>supported                       |
|                    |       |       |                   | 0 – Unsupported                           |
|                    |       |       |                   | 1 – Supported                             |
|                    | 14    | 1     | FRESETS           | TRESET<br>supported                       |
|                    |       |       |                   | 0 – Unsupported                           |
|                    |       |       |                   | 1 – Supported                             |
|                    | 17:15 | 3     | PWRMODES[2:<br>0] | PWRMODES[x]<br>– Power Control<br>Mode    |
|                    |       |       |                   | 0 – Unsupported                           |
|                    |       |       |                   | 1 – Supported                             |
|                    | 18    | 1     | PCMR              | PCMR – PM<br>default<br>POWRMODE<br>value |
|                    |       |       |                   | 0 – No                                    |
|                    |       |       |                   | 1 – Yes                                   |

Table continues on the next page...

Table 49-11. Configuration Register 0 Format (continued)

| Field         | Bit # | Width | Name        | Description                              |
|---------------|-------|-------|-------------|------------------------------------------|
| T2<br>Options | 19    | 1     | DCAS        | DCAS – Decouple at start-up              |
|               |       |       |             | 0 – No                                   |
|               |       |       |             | 1 – Yes                                  |
| T4<br>Options | 21:20 | 2     | SSCANS[1:0] | x0 – SScan1:0 Unsupported                |
|               |       |       |             | x1 – SScan1:0 Supported                  |
|               |       |       |             | 0x – SScan3:2 Unsupported                |
|               |       |       |             | 1x – SScan3:2 Supported                  |
|               | 24:22 | 3     | OSCANS[2:0] | xx0 – OScan3:2 Unsupported               |
|               |       |       |             | xx1 – OScan3:2 Supported                 |
|               |       |       |             | x0x – OScan5:4 Unsupported               |
|               |       |       |             | x1x – OScan5:4 Supported                 |
|               |       |       |             | 0xx – OScan7:6 Unsupported               |
|               |       |       |             | 1xx – OScan7:6 Supported                 |
|               | 25    | 1     | APFCS       | Auxiliary Pin Function Control supported |
|               |       |       |             | 0 – Unsupported                          |
|               |       |       |             | 1 – Supported                            |
|               | 26    | 1     | TAPWIDS     | TAP Width Default                        |
|               |       |       |             | 0 – 2-pin supported                      |
|               |       |       |             | 1 – 4-pin supported                      |
|               | 27    | 1     | TAPWLCK     | TAP Width Locked                         |
|               |       |       |             | 0 – Not locked                           |
|               |       |       |             | 1 – Locked                               |
|               | 28    | 1     | STCKDCS     | SYS_TCK Duty Cycle supported             |
|               |       |       |             | 0 – Unsupported                          |
|               |       |       |             | 1 – Supported                            |

Table continues on the next page...



**Table 49-11. Configuration Register 0 Format (continued)**

| Field    | Bit # | Width | Name     | Description            |
|----------|-------|-------|----------|------------------------|
| Reserved | 31:29 | 3     | Reserved | Reserved, read as zero |

**49.7.2.3.15 Register reset values****Table 49-12. Reset values of TAP.7 controller registers managed with commands**

| Register type    | Width | Register mnemonic | Name                      | Values for reset type |       |     |     |
|------------------|-------|-------------------|---------------------------|-----------------------|-------|-----|-----|
|                  |       |                   |                           | 0                     | 1 – 3 | 4   | 5   |
| <b>Control</b>   | 1     | ECL               | Exit Control Level        | 0                     | 0     | 0   | NC  |
|                  | 1     | SUSPEND           | Suspend                   | 0                     | 0     | 0   |     |
|                  | 1     | ZBSINH            | ZBS Detect Inhibit        | 0                     | 0     | 0   |     |
|                  | 5     | SCNFMT            | Scan Format               | DF                    | DF    | DF  |     |
|                  | 1     | SSDE              | SSD Enable                | 0                     | 0     | 0   |     |
|                  | 2     | DLYC              | Delay control             | 00                    | 00    | 00  |     |
|                  | 2     | RDYC              | Ready Control             | 00                    | 00    | 00  |     |
| <b>Options</b>   | 1     | TRESET            | Test Reset                | 0                     | 0     | 0   | NC  |
|                  | 2     | APFC              | Aux. Pin Functional Cntl. | DP                    | DP    | NC  |     |
| <b>Select</b>    | 1     | CGM               | Conditional Group Member  | 0                     | 0     | 0   | NC  |
|                  | 1     | SGC               | Scan Group Candidate      | DCA                   | DCA   | DCA |     |
|                  | 1     | SREDGE            | Sampling Rising Edge      | 0                     | 0     | 0   |     |
| <b>Read-only</b> | 32    | RDBACK0           | Read-back 0               | N/A                   | N/A   | N/A | N/A |
|                  | 32    | CNFG0             | Configuration Register 0  |                       |       |     |     |

NC = No Change

DF = Default Scan Format

DM = Default Power Control Mode

DCA = Default Coupling Action

DT = Default Topology

DP = Default Pin Function

**49.7.3 EPU Commands**

There are two command types:

- Two Part Commands
- Three Part Commands

Any number of two-part and three part commands may be issued in any combination before the command control level is exited. If the command control level is exited before a two- or three-part command is completed, the command is aborted and becomes a NOP.

### 49.7.3.1 Two Part Commands

TAP.7 controller commands are 10-bit values. The 10-bit commands are created by two consecutive DR-Scans when the control level is locked at the command control level (two). The first DR-Scan provides a five-bit command opcode. The second DR-Scan provides a five-bit operand. These two scans are called the command part one (CP1) and command part two (CP2), respectively. A command created entirely with these two scans is called a two-part command.

### 49.7.3.2 Three Part Commands

Some commands are used to send and/or receive data values other than values embedded in the command's operand. These commands are called three-part commands. With these commands, CP1 and CP2 are followed by an additional DR-Scan to transport a data value to or from an EPU scan path. When used for this purpose, the DR-Scan is called a Control Register Scan (CR-Scan). The CR-Scan is a minimum of zero bits in length with there being no maximum length. The CR-Scan path is described in SECTION.

### 49.7.3.3 Command Sequence

The two 5-bit values concatenated to create the 10-bit command represent the number of Shift-DR states between the *Capture-DR* and *Update-DR* states of CP1 and CP2. The count within CP1 creates the MSBs of the command while the count within CP2 creates the LSBs of the command. The command is decoded when the *Update-DR* state of the second DR-Scan is reached. A determination as to whether the command is a two- or three-part command is made at this point. The command format is shown in the following figure.

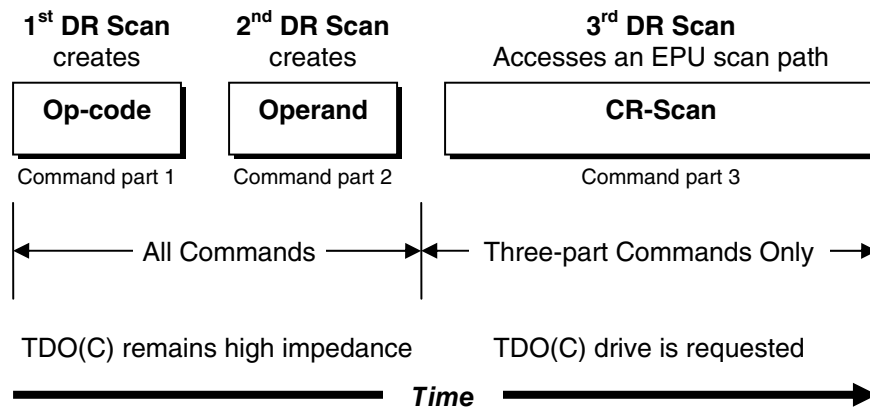


**Figure 49-8. DR\_Scan sequence used for command creation**

The 10-bit command created by CP1 and CP2 determines the command's function and whether the command is a two- or three-part command. If the command is a two part command, the function specified by the command is performed when command part 2 completes. If the command is a three-part command, the DR-Scan following CP2 performs a TAP.7 controller function designated by the 10-bit controller command. This DR-Scan is called a **Control-Register Scan** (CR-Scan) and can be any length. A CR-Scan has many of the attributes of the DR-Scan as it moves data between the external tool and TAP.7 controller and accesses various EPU scan paths. There are many two part commands, but only three three-part commands, each having a special purpose.

### 49.7.3.4 Commands

There are 11 mandatory and 5 optional commands that can be grouped in classes as shown in the following list:

- Store – The operand is stored in a register or causes an action.
- Select – Controls participation of Scan and conditional command execution.
- Scan – Inputs and outputs with a CR-Scan.
- Enumerate – Controller ID allocation and de-allocation.
- Private – Commands available for chip-specific definition.

The definition of the commands with their operands is shown in the following table.

**Table 49-13. – TAP.7 controller command list**

| Command class          | Opcode count | Mnemonic | Name                        | TAP.7 class |   |   |   |
|------------------------|--------------|----------|-----------------------------|-------------|---|---|---|
|                        |              |          |                             | 1           | 2 | 3 | 4 |
| Store                  | 0x00         | STMC     | Store Miscellaneous Control | M           | M | M | M |
|                        | 0x01         | STC1     | Store Conditional 1-bit     | M           | M | M | M |
|                        | 0x02         | STC2     | Store Conditional 2-bit     | M           | M | M | M |
|                        | 0x03         | STFMT    | Store Format                | -           | M | M | M |
|                        | 0x04         | STDCST   | Store Data Channel State    | -           | - | - | - |
| Select                 | 0x06         | MCM      | Make Cond. Group Member     | M           | M | M | M |
|                        | 0x07         | MSC      | Make Scan Group Candidate   | -           | M | M | M |
| Scan                   | 0x08         | SCNB     | Scan Bit                    | M           | M | M | M |
|                        | 0x09         | SCNS     | Scan String                 | M           | M | M | M |
| Enumerate <sup>1</sup> | 0x0A         | CIDA     | Allocate Controller ID      | -           | - | M | M |
|                        | 0x0B         | CIDD     | De-allocate Controller ID   | -           | - | M | M |
| Reserved               | 0x0C–0x1F    | Reserved | Reserved                    | -           | - | - | - |

1. Enumerate commands are not currently supported by the CJTAG module.

#### 49.7.3.4.1 Store Commands

Store commands are two-part commands that are available in both Star and Series Scan Topologies. The Store Miscellaneous Control (STMC) Command performs control functions that affect all TAP.7 controllers sharing an external tool connection. It stores values in one-bit and two-bit registers.

The Store Conditional 1 bit (STC1) and Store Conditional 2 bit (STC2) Commands store one-bit and two-bit values in registers, respectively. These commands operate two ways:

1. Unconditionally storing registers in all TAP.7 controllers or
2. Storing registers in only the TAP.7 controllers that have a logic 1 Conditional Group Member (CGM) Register value.

The Store Format (STFMT) Command unconditionally stores a five-bit Scan Format value.

The Store Data Channel State (STDCST) Command stores a five-bit value identifying the states supporting transport.

### 49.7.3.4.2 Select Commands

Select commands are two-part commands that are available in both Star and Series Scan Topologies. Their primary use is to select TAP.7 controllers of interest in a Star Scan Topology. These commands define the TAP.7 controller(s) participating in scans and the TAP.7 controllers that conditionally execute STC1 and STC2 Commands.

The Make Conditional Group Member (MCM) Command specifies which TAP.7 controllers execute store conditional commands as it manages the Command and Path Enable (CAPE) Register. This register also affects the execution of private commands and is utilized in EPU scan path selection.

The Make Scan Group Candidate (MSC) command manages the SGC register. This register determines whether the STL is coupled when the *Run-Test/Idle* state is reached when scan formats other than JScan0 and JScan1 are used. Select commands execute only when the CIDI Register is a logic 0, indicating the TAP.7 controller's CID is valid, and are treated as no operations otherwise.

### 49.7.3.4.3 Scan Commands

Scan commands are three-part commands that are available in both Star and Series Scan Topologies. The CR-Scan portion of Scan Bit (SCNB) and Scan String (SCNS) Commands is used to set and test bits (SCNB) or transfer strings of bits (SCNS). The commands may be used to access both public and private registers.

## 49.7.3.5 Command Definitions

### 49.7.3.5.1 Store Commands

#### 49.7.3.5.1.1 Store Miscellaneous Control Command (STMC)

Table 49-14. Store Miscellaneous Control

| STMC   |         | Store Miscellaneous Control |          |   |   |                            |
|--------|---------|-----------------------------|----------|---|---|----------------------------|
| Opcode | Operand | bbb                         |          | x | y | Description                |
| 00000  | bbb x y | 000                         | StateCtl | 0 | 0 | NOP                        |
|        |         |                             |          | 0 | 1 | ExitCmdLev(ECL)            |
|        |         |                             |          | 1 | 0 | Exit/Suspend (SUSPEND) = 1 |
|        |         |                             |          | 1 | 1 | ZBS Inhibit (ZBSINH) = 1   |
|        |         | 001                         | ScanCtl  | 0 | - | SGC = y                    |
|        |         |                             |          | 1 | - | CGM = y                    |

Table continues on the next page...

**Table 49-14. Store Miscellaneous Control (continued)**

| STMC   |         | Store Miscellaneous Control |          |   |   |                      |
|--------|---------|-----------------------------|----------|---|---|----------------------|
| Opcode | Operand | bbb                         |          | x | y | Description          |
|        |         | 010                         | RdyCtl   | - | - | RDYC = xy            |
|        |         | 011                         | DlyCtl   | - | - | DLYC = xy            |
|        |         | 100                         | ScnFunc  | 0 | - | SSDE = y             |
|        |         |                             |          | 1 | 0 | Star-4 Topology Test |
|        |         |                             |          | 1 | 1 | Reserved             |
|        |         | 101                         | Reserved | - | - | Reserved             |
|        |         | 110                         |          |   |   |                      |
|        |         | 111                         |          |   |   |                      |

**49.7.3.5.1.2 Store Conditional 1-bit (STC1)****Table 49-15. Store Conditional 1 bit (STC1)**

| STC1   |         | Store Conditional 1 bit |           |      |             |
|--------|---------|-------------------------|-----------|------|-------------|
| Opcode | Operand | c                       | bbb       | CAPE | Description |
| 00001  | c bbb v | 0                       | 000       | -    | SREDGE = v  |
|        |         |                         | 001       | -    | Reserved    |
|        |         |                         | 010       | -    | TRESET = v  |
|        |         |                         | 011 – 111 | -    | Reserved    |
|        |         | 1                       | —         | 0    | No change   |
|        |         |                         | 000       | 1    | SREDGE = v  |
|        |         |                         | 001       | 1    | Reserved    |
|        |         |                         | 010       | 1    | TRESET = v  |
|        |         |                         | 011 – 111 | 1    | Reserved    |
|        |         |                         |           |      |             |

**49.7.3.5.1.3 Store Conditional 2-bit (STC2)****Table 49-16. Store Conditional 2 bit (STC2)**

| STC2   |         | Store Conditional 2 bit |    |      |             |
|--------|---------|-------------------------|----|------|-------------|
| Opcode | Operand | c                       | bb | CAPE | Description |
| 00010  | c bb vv | 0                       | 00 | -    | Reserved    |
|        |         |                         | 01 | -    | STCKDC = vv |
|        |         |                         | 10 | -    | APFC = vv   |
|        |         |                         | 11 | -    | Reserved    |
|        |         | 1                       | —  | 0    | No change   |
|        |         |                         | 00 | 1    | Reserved    |
|        |         |                         | 01 | 1    | STCKDC = vv |

Table continues on the next page...

**Table 49-16. Store Conditional 2 bit (STC2) (continued)**

| STC2   |         | Store Conditional 2 bit |    |      |             |
|--------|---------|-------------------------|----|------|-------------|
| Opcode | Operand | c                       | bb | CAPE | Description |
|        |         |                         | 10 | 1    | APFC = vv   |
|        |         |                         | 11 | 1    | Reserved    |

**49.7.3.5.1.4 Store Scan Format (STFMT)****Table 49-17. Store Scan Format (STFMT)**

| STFMT  |         | Store Scan Format |
|--------|---------|-------------------|
| Opcode | Operand |                   |
| 00011  | nnnnn   | SCNFMT = nnnnn    |

**49.7.3.5.2 Select Commands**

This section describes the select commands with their operands.

**49.7.3.5.2.1 Make Conditional Group Member (MCM)**

This command operates as a no operation when the CIDI Register is a logic 1 (The Controllers CID is invalid). Otherwise it sets the CGM bit in a TAP.7 controller that has a valid CID and the CID matches the operand CID field (iiii). The CGM Register is also stored with the STMC and SCNB commands.

**Table 49-18. Make Conditional Group Member (MCM)**

| MCM    |         | Conditional Group Member |                                                                                                   |
|--------|---------|--------------------------|---------------------------------------------------------------------------------------------------|
| Opcode | Operand | m                        | Description                                                                                       |
| 00110  | m iiii  | 0                        | CGM bit of the non-targeted controller is cleared                                                 |
|        |         | 1                        | CGM bit of the targeted controller is set.<br>CGM bit of a non-targeted controller is unaffected. |

**49.7.3.5.2.2 Make Scan Group Candidate (MSC)**

This command operates as a no operation when the CIDI Register is a logic 1 (The Controllers CID is invalid). Otherwise it sets the SGC bit in a TAP.7 controller that has a valid CID and the CID matches the operand CID field (iiii). The SGC Register is also stored with the STMC and SCNB commands.

**Table 49-19. Make Scan Group Candidate (MSC)**

| MSC    |         | Make Scan Group Candidate |                                                                                                   |
|--------|---------|---------------------------|---------------------------------------------------------------------------------------------------|
| Opcode | Operand | m                         | Description                                                                                       |
| 00111  | m iiiii | 0                         | SGC bit of the non-targeted controller is cleared                                                 |
|        |         | 1                         | SGC bit of the targeted controller is set.<br>SGC bit of a non-targeted controller is unaffected. |

### 49.7.3.5.3 Scan Commands

#### 49.7.3.5.3.1 Scan Bit (SCNB)

This is a 3-part command. The operand chooses the bit to be read or written by the CR-Scan.

**Table 49-20. Scan Bit (SCNB)**

| SCNB   |         |         | Scan Bit |                                                        |            |
|--------|---------|---------|----------|--------------------------------------------------------|------------|
| Opcode | Operand | CR-Scan | yyyyy    | Description                                            | R/W        |
| 01000  | yyyyy   | -       | 0        | Scan Group Candidate (SGC)                             | Write only |
|        |         | -       | 1        | Conditional Group Member (CGM)                         | Write only |
|        |         | -       | 2        | CNFG[0]                                                | Read only  |
|        |         | -       | 3        | CNFG[1] (not implemented)                              | Read only  |
|        |         | -       | 4        | CNFG[2] (not implemented)                              | Read only  |
|        |         | -       | 5        | CNFG[3] (not implemented)                              | Read only  |
|        |         | -       | 6        | Functional reset requested (FRESETR) (not implemented) | Read only  |
|        |         | -       | 7        | Series topology detect                                 | Write only |
|        |         | -       | 15 – 8   | Reserved                                               | Read only  |

#### 49.7.3.5.3.2 Scan String (SCNS)

This is a 3-part command. The operand chooses the string to be read or written by the CR-Scan.

**Table 49-21. Scan String (SCNS)**

| SCNS   |         |         | Scan String |                           |           |
|--------|---------|---------|-------------|---------------------------|-----------|
| Opcode | Operand | CR-Scan | yyyyy       | Description               | R/W       |
| 01001  | yyyyy   | -       | 0           | RDBACK[0] Register [31:0] | Read only |

*Table continues on the next page...*

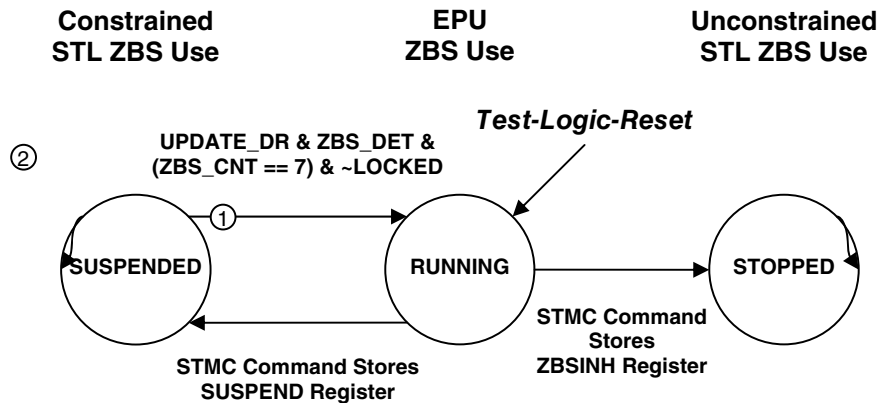


**Table 49-21. Scan String (SCNS)**  
(continued)

| SCNS   |         |         | Scan String |                                             |           |
|--------|---------|---------|-------------|---------------------------------------------|-----------|
| Opcode | Operand | CR-Scan | yyyyy       | Description                                 | R/W       |
|        |         | -       | 1           | RDBACK[1] Register [31:0] (not implemented) | Read only |
|        |         | -       | 2           | CNFG[0] Register [31:0]                     | Read only |
|        |         | -       | 3           | CNFG[1] Register [31:0] (not implemented)   | Read only |
|        |         | -       | 4           | CNFG[2] Register [31:0] (not implemented)   | Read only |
|        |         | -       | 5           | CNFG[3] Register [31:0] (not implemented)   | Read only |
|        |         | -       | 6 - 7       | Reserved (read logic 0)                     | Read only |
|        |         | -       | 31 - 8      | Private – available for customization       |           |

#### 49.7.4 EPU operating states

The TAP.7 controller and STL share the use of ZBSs. The EPU has three operating states as shown in the following figure:

**Figure 49-9. EPU operating states**

The values of the SUSPEND and ZBSINH Registers determine the type of operation as shown in the following table. The Test-Logic-Reset state sets the SUSPEND and ZBSINH Register values to a logic 0.

**Table 49-22. ZBS use state characteristics with both TAP.7 and other ZBS use**

| ZBSINH | SUSPEND | EPU operating State | ZBS use by: | Until                                   |
|--------|---------|---------------------|-------------|-----------------------------------------|
| 0      | 0       | RUNNING             | EPU         | Command allocates to STL                |
| 0      | 1       | SUSPENDED           | STL         | ZBS sequence allocates to EPU           |
| 1      | 0       | STOPPED             | STL         | Test-Logic-Reset state allocates to EPU |
| 1      | 1       | Not possible        |             |                                         |

### 49.7.5 System and EPU Paths

The System path is utilized when the external tool requires access to system (STL) resources. This path is used when either the SUSPEND Register is a logic 1 or the ZBS count is less than or equal to one. The EPU path is used otherwise.

The EPU scan path is constructed from five types of EPU scan paths:

- **Bypass** – A 1-bit scan path
- **Bit** – Supports set and test bit operations (a 1-bit shift register)
- **String** – Support 32-bit reads or 32-bit writes
- **Enumerate** – A 36-bit scan path that transports a TAPC address with T3 and above TAP.7s
- **Auxiliary** – An n-bit scan path that transports 1 – n bits using a conventional IEEE 1149.1 style DR-Scan path using Control Levels 4 and 5.

## 49.8 APU (Advanced Protocol Unit) Operation

### 49.8.1 Overview

The Advanced Protocol Unit (APU) is placed between the EPU and the RSU to create the T4 TAP.7 controller. The APU delivers advanced capabilities with a number of operating modes (new scan formats) that utilize the Standard Protocol and the Advanced Protocol in both the narrow and wide T4 TAP.7 versions.

The APU enables scan transfers with two signals by serializing the information related to a TAPC state machine state. It provides a number of serialization formats to balance scan flexibility and scan performance. It multiplexes the use of the TMS signal for T4 TAP.7 controller functions. The serialization is bypassed to provide T3 TAP.7 capability.

With a wide TAP.7, the chip architect may choose to implement the TDIC and TDOC signal functions as fixed functionality (the T3 TDIC and TDOC functions) or as programmable between the T3 TDIC and TDOC signal functions and an alternate function using a TAP.7 controller register. In this case, the default signal function may be either of the programmable choices. Programmable TDIC and TDOC functionality provides a means to fully utilize these signals in any scan topology as the T3 TAP.7 functionality is fully supported.

The entire EPU infrastructure (i.e., ZBS detection, control-level management, and command generation) is utilized when either the Standard Protocol or the Advanced Protocol is used, as these TAP.7 controller attributes are controlled entirely from the TAPC state machine state progression. The EPU is unaware of the APU's existence. All T4 features are managed outside of the EPU.

The TAP.7 serialization schemes are specified with scan formats that complement the JScan 0-3 Scan Formats inherited from the T3 TAP.7. The scan formats are called advanced scan formats as they support two-signal operation.

Two groups of scan formats (MScan and OScan0-7) provide a number of options for the serialization of scan information for use in a Star-2 scan topology. The OScan Scan Format is broken into two groups with the same function, one group has better performance but may only be used with an external tool sourced TCKC. The external tool chooses a scan format that matches the constraints imposed by the application, chip components, and possibly the external tool itself.

The MScan and OScan formats address Test and Debug use cases. The mandatory MScan and OScan0-1 formats emphasize flexibility over performance. The remaining optional Oscan2-7 formats emphasize performance over flexibility by not transmitting unneeded information such as TDI and TDO in non-shift TAPC state machine states.

The optional capabilities added with T4 and above TAP.7s change the bit sequences seen at the TAP.7 signals. A TAP.7 controller comprehends only the bit sequences generated by the use of the features it supports. It is never exposed to bit sequences it does not comprehend. A check for supporting an enabled feature is made before the feature is used (alters the bit sequences). When a T4 TAP.7 controller detects the use of an unsupported scan format, it halts its operation and places itself in a state where its operation may be resumed. This state is henceforth called "offline". A TAP.7 controller that is not offline is henceforth called "online". While offline the TAP.7 controller awaits the detection of a

special signaling sequence called the Online Activation Protocol. This protocol sequence places an offline TAP.7 controller online and synchronizes the operation of all online TAP.7s.

The Advanced Protocol Unit supports the stall of a transfer by both the external tool and STL at any TAPC state.

The APU provides the following functions:

- Bypass (for operation with the Standard Protocol)
- Scan control (for SP use)
- Configuration Control (for CP use)

The Scan Control, Configuration Control, and Transport Control functions are idled when the Standard Protocol is used as the Bypass Function connects the TCKC and TMSC pins to the EPU.

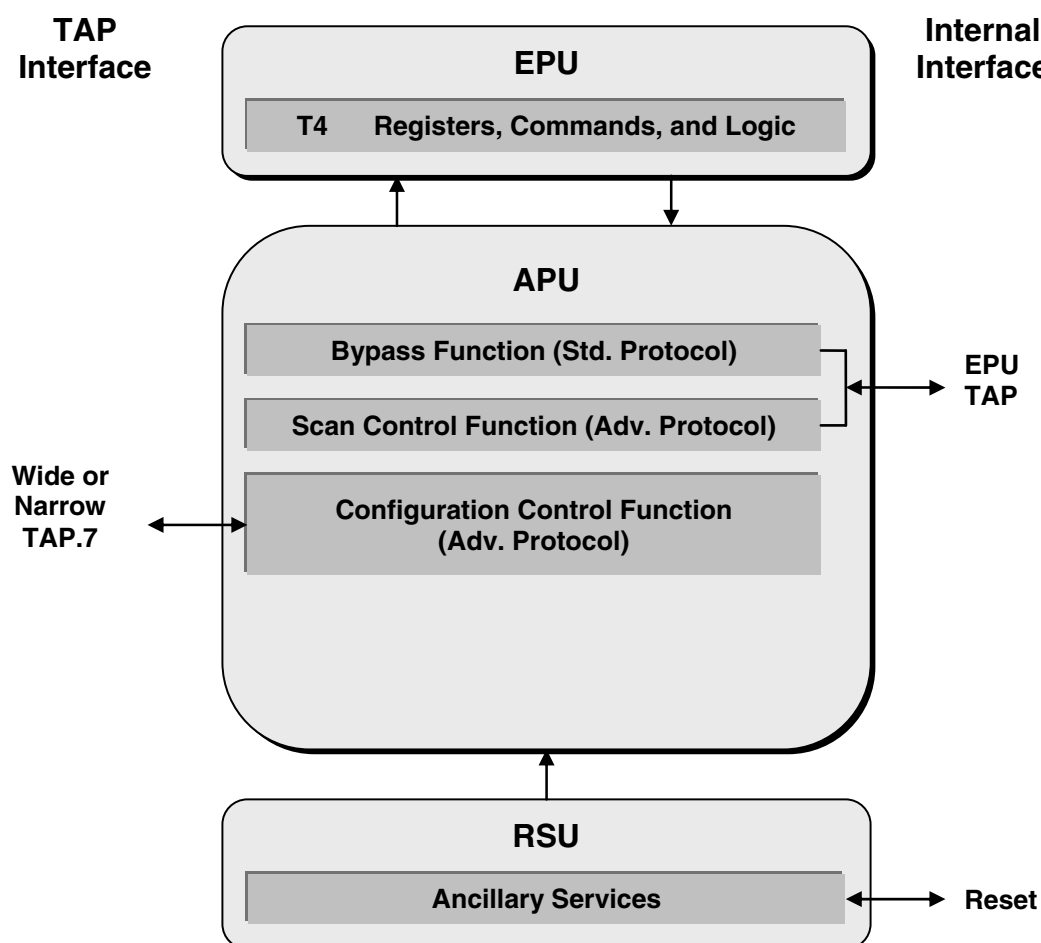


**Figure 49-10. Conceptual block diagram of the APU functions**

The APU's internal interface includes a connection to EPU registers, the EPU's TAP interface, connections to chip-level data channels (when they are implemented), and reset signaling with the EPU. The APU's TAP interface provides mutually exclusive use of the TMS pin for:

- Bypass Function
- Scan Control Function
- Configuration Control Function

When the Standard Protocol is used, the APU's Bypass function connects the EPU directly to the TAP.7 pins. When the TDIC and TDOC pin functions are not available (narrow TAP.7 or TDIC and TDOC pin functions are an alternate function), the APU sources a logic 1 to the EPU inputs associated with these pins.

When the Advanced Protocol is used, the APU multiplexes the use of the TMS pin between the other APU functions. Scan Control acts as a serial to parallel converter, translating the parallel information at the EPU's TAP interface into bidirectional serial transfers. Since the Scan Control is literally a protocol adaptor, the entire EPU infrastructure (ZBS detection, control-level management, and command generation) is utilized when either the Standard or Advanced Protocol is used, as these TAP.7 controller attributes are controlled entirely from the TAPC state progression.

## 49.8.2 Operation

The conceptual view of the APU operation is shown in the following figure.

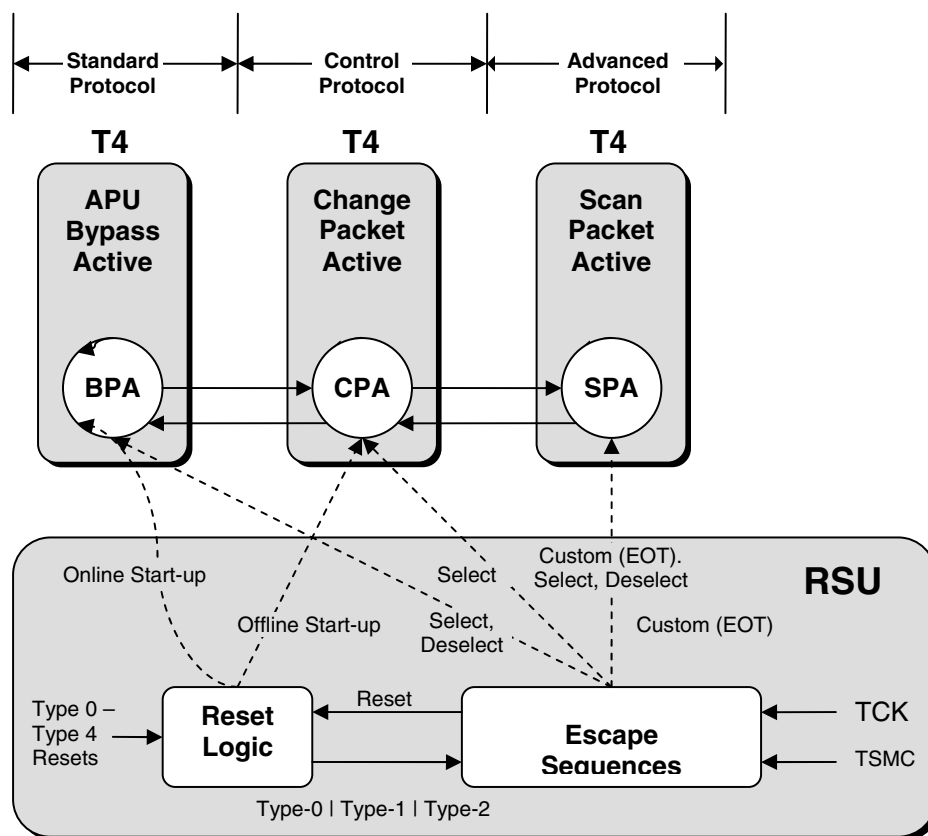


Figure 49-11. Conceptual view of APU operation

The APU operating states managing the use of the TAP pins and the ancillary (RSU) services use of the TMS pin.

APU operating states:

- **BPA** – **By**Pass **A**ctive
- **CPA** – **C**hange **P**acket **A**ctive
- **SPA** – **S**can **P**acket **A**ctive

#### 49.8.2.1 BPA

The *BPA* state allocates the use of the TMS pin to the Standard Protocol. This state is entered two ways:

- As a result of a TAP.7 controller reset when the TAP.7 controller starts up online.
- From the CPA state when a TAP.7 controller register write specifies the use of the Standard Protocol.

### 49.8.2.2 CPA

The *CPA* state allocates the use of the TMS pin to a CP. This state is entered in one of three ways:

- As a result of a TAP.7 controller reset when the TAP.7 controller starts up offline.
- A value specifying the use of the Advanced Protocol is written to the SCNFMT Register when using the Standard Protocol.
- A TAP.7 controller register write occurs when using the Advanced Protocol.

### 49.8.2.3 SPA

The *SPA* state allocates the use of the TMS pin to an SP. This state is entered when the SCNFMT Register specifies the use of the Advance Protocol when a CP ends.

## 49.8.3 Escape sequences

Escape-sequence detection provides services to the Reset Logic, Scan, Configuration Control, and Transport Control functions as outlined below:

- **Scan and Transport Control** – End of Transfer (EOT) escape sequence detected.
- **Configuration Control** – Synchronize Advanced Protocol (SAP) escape sequence.
- **Reset Logic** – Reset (RES) escape sequence detected.

An EOT escape sequence is used to increase the efficiency of OScan Scan formats.

An SAP escape sequence is used to place TAP.7 controllers online.

A RES escape sequence asynchronously causes generation of a Type-3 Reset. Following a RES escape sequence, the drive of the TMS pin is inhibited so it is not driven when TCKC becomes a logic 0 immediately following the RES escape sequence.

## 49.8.4 Signal behaviors

The TCKC signal:

- The test clock

The TMS signal:

- An input when the Standard Protocol is used
- Bi-directional when the Advanced Protocol is used

The TDI and TDO signals:

- Deleted with a narrow TAP.7
- One of two functions with a wide TAP.7
  - Fixed – as the TAP.7 TDIC and TDOC pin functions, or
  - Programmable – as the TAP.7 TDIC and TDOC signal functions or alternate functions with a TAP.7 controller reset establishing the default signal function.

## 49.8.5 APU Functions

### 49.8.5.1 Bypass Active (BPA) Function

When the Standard Protocol is used, the APU connects the EPU inputs and outputs directly to the TAP.7 pins. If the TDIC and TDOC pin functions are not present (e.g., a narrow TAP.7 or the TDIC and TDOC pins have an alternate function assigned to them), the APU provides a logic 1 to the EPU for unsupported pin functions and ignores the EPU's TDO output data.

### 49.8.5.2 Scan Packet Active (SPA) Function

The Scan Control function is encapsulated within the Scan Packet Active (SPA) state. With the Standard Protocol, the external tool and TAP.7 exchange the scan information associated with a single TAPC state machine state in one TCK period using the TMS(C), TDI(C), and TDO(C) pins. With the Advanced Protocol all or part of this information is exchanged serially within a Scan Packet (SP) as shown in . There is a one to one correspondence between SPs and TAPC state machine state changes, except when the SPA state precedes the CPA state. In this case the SP preceding a CP does not advance the TAPC state machine state.



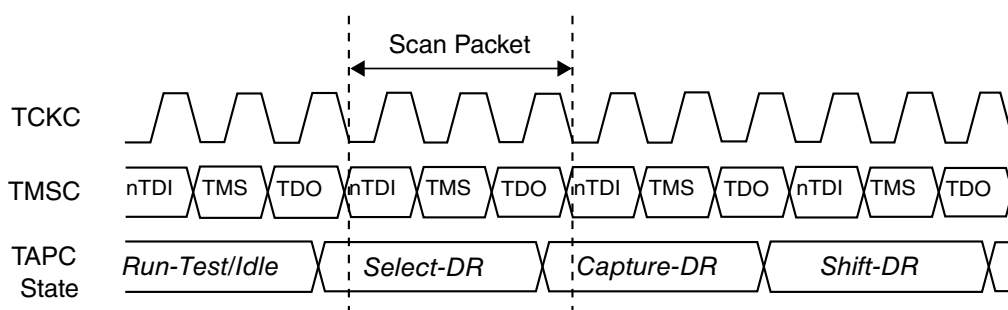


**Figure 49-12. Scan Packet Sequence**

A comparison of the parallel transfer of TAP information with the Standard Protocol and the serialization and de-serialization of the TAP information using the TCKC and TMSC pins with the Advanced Protocol is shown in FIGURE. The IEEE 1149.1 signals between the EPU and STL are prefixed with SYS\_.

The APU manages the serialization and de-serialization of the scan information. The APU converts the TDI and TMS information within a Scan Packet (SP) into the TMS and TDI information presented to the EPU. It converts the combination of the EPU TDO and EPU TDO\_OE signals into the TDO information in the same SP. If the EPU indicates that the TDO pin drive is not enabled, logic 1 TDO data is created, otherwise the TDO data is the supplied by either the EPU or STL.

Information within a Scan Packet is first passed from the external tool to the TAP.7 followed by information passed from the TAP.7 to the external tool. In this example the scan format specifies the exchange of the TMS, TDI, and TDO TAPC information. Each TAPC state machine state takes three TCKC periods. Other scan formats specify the exchange of different mixes of information. Control information is added with the most flexible scan formats. Some scan formats send less information and consequently take fewer TCKC periods.

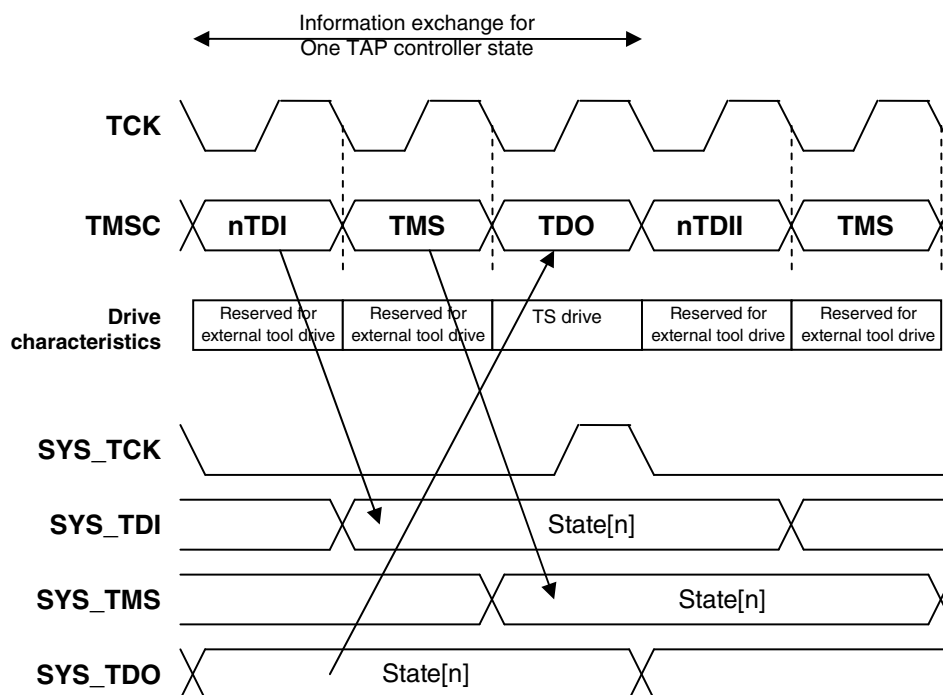


**Figure 49-13. Serialization and De-serialization of Scan Packet Information**

In order to produce comparable TAPC state machine state rates with a TAP.1 and TAP.7 controller:

- Full-cycle timing between the external tool and TAP.7 allows operation at higher TCKC frequencies.
- The number of bits transferred to advance a TAPC state machine is minimized. (reaching as few as one for some scan formats.)

The TAP.7 controller provides both full-cycle bit timing (falling TCKC edge to falling TCKC edge) and half-cycle bit timing (falling TCKC edge to rising TCKC edge), with a TAP.7 register used to define the type of timing used. Half-cycle timing is used as the default following a TAP.7 controller reset. The TCKC frequency is reduced to accommodate half-cycle timing.

With full-cycle timing, the TMSC input and output data are both output with and sampled with the falling edge of the TCKC. This means that the entire TCKC period may be spent transferring a bit from the external tool to the TAP.7 or vice-versa. The TCKC frequency can therefore be maximized when full-cycle timing is used. With half-cycle timing, the APU samples the TMSC input data with the rising edge of the TCKC and outputs TMSC data with the falling edge of TCKC.

Minimizing the number of bits transferred per TAPC state machine state is affected by a number of factors. The information that must be transferred per TAPC state machine state varies for different test and debug applications. Also, with some test and debug

applications, all TAPC state machine states do not require that the same amount of information be transferred. This is especially true with debug applications where specialized use of the TAP is more likely.

### 49.8.5.3 Control Process Active (CPA) Function

While using the Advanced Protocol, all TAP.7 configuration changes occur within the CPA state. All online and offline transitions also occur while in the CPA state. When entering the CPA state from the BPA or SPA state, the CPA state checks for the use of supported features affecting the Advanced Protocol. The CPA state is exited when the enabled features affecting the Advanced Protocol are supported.

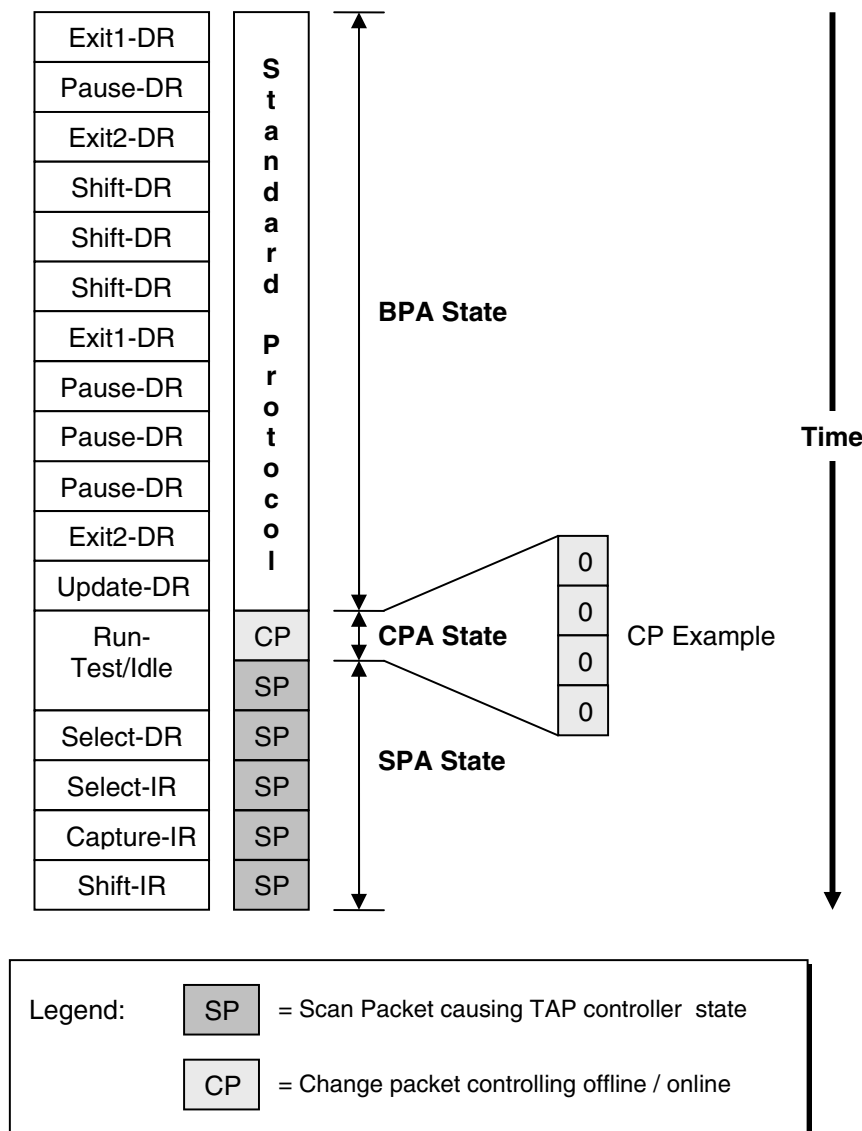


**Figure 49-14. CPA State Entry from BPA State with Continued Online Operation**

An SP preceding a CP does not advance the TAPC state machine state. This SP performs housekeeping functions that assure the contents of the SP preceding this SP have been fully utilized. This has significance when the SP contains control information that allows the STL to stall the completion of a SP.

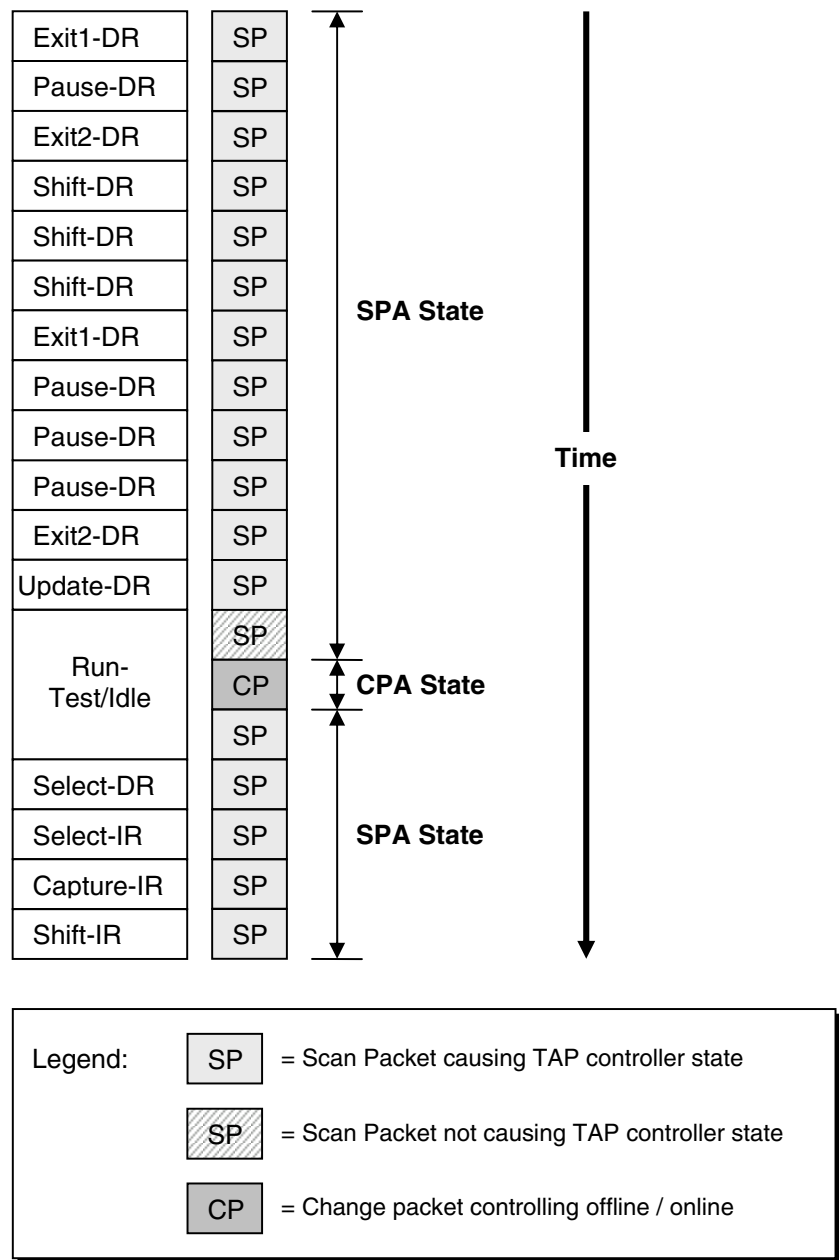


Figure 49-15. CPA State Entry from the SPA State with Continued Online Operation

When an unsupported feature is enabled, the TAP.7 controller is placed offline. This suspends the TAP.7 controller's operation at a point where its operation may later be synchronized to the remainder of the system. The APU stops advancing the EPU's and CLTAPC's TAPC state machine state while offline. It also suspends Data Channels activity while offline. The detection of the appropriate Online Activation Protocol while

the TAP.7 controller is offline places the TAP.7 controller online and causes an exit from the CPA state to the SPA state. An offline to online transition initiates the use of a predefined set of mandatory features in all online TAP.7 controllers to assure they operate synchronously.

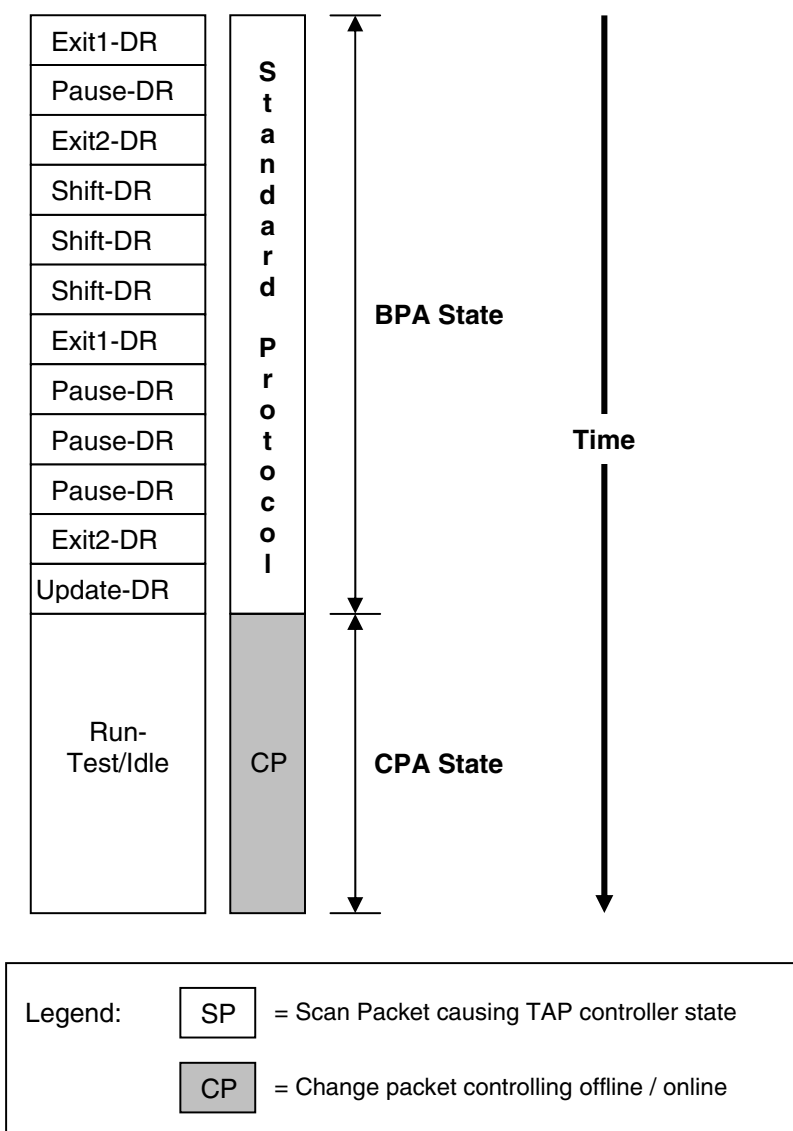


**Figure 49-16. CPA State Entry from the BPA State with Offline Operation**

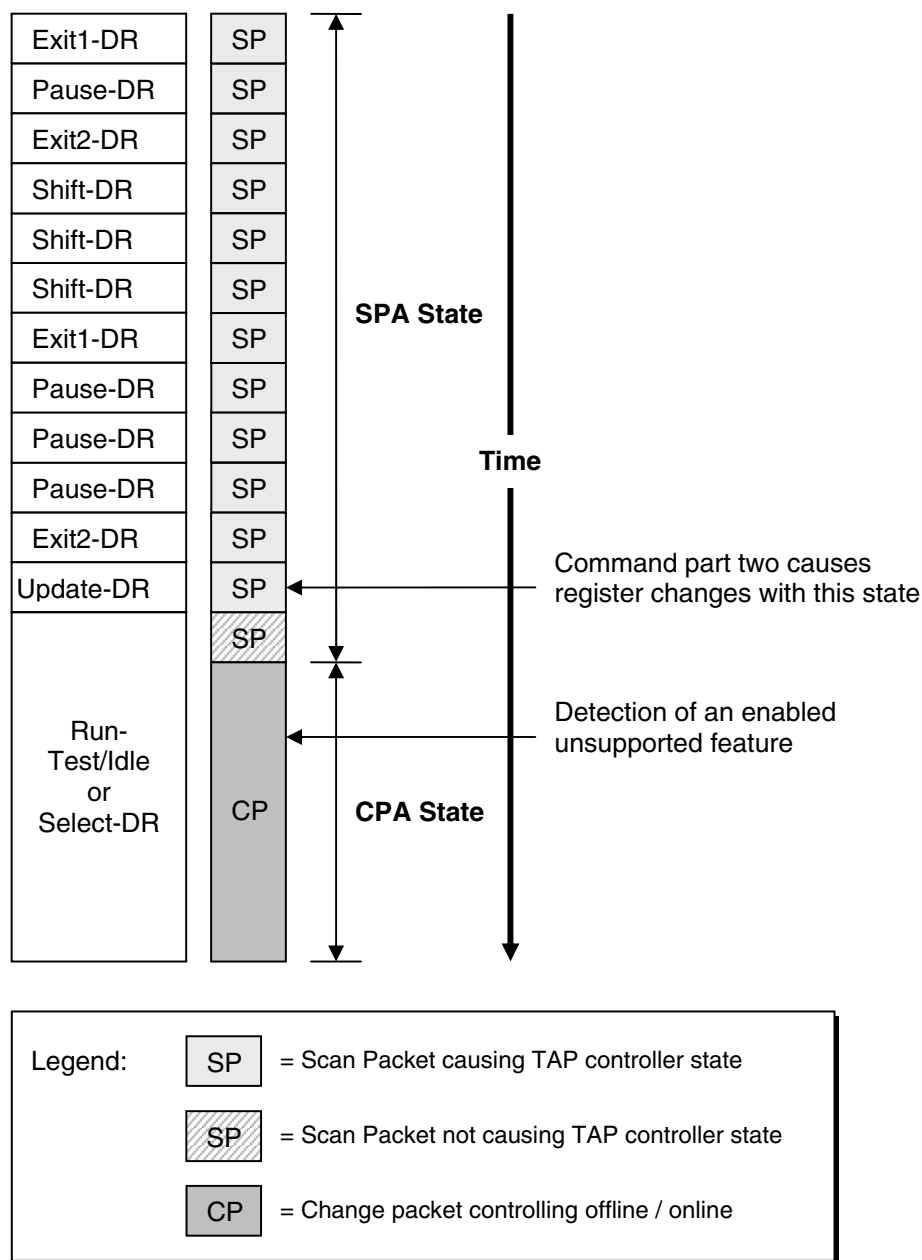


**Figure 49-17. CPA State Entry from the SPA State with Offline Operation**

A TAP.7 controller reset places the TAP.7 controller offline when the offline at start-up option is used. From this point the TAP.7 controller awaits being placed online.

## 49.8.6 Configuration Change Packets (CP)

A Change Packet (CP) is composed of three element types:

- **Preamble** – A 1-bit value that is a don't care. This bit facilitates a Standard to Advanced Protocol change with some external tool implementations.

- **Body** – Two or more bits representing one or more CP directives
  - **00** – CP\_END
  - **01** – CP\_NOP (extends the body by 1-bit)
  - **10** – CP\_NOP (extends the body by 1-bit)
  - **11** – CP\_RES (Generate a Type-3 TAP.7 controller reset coincident with the postamble bit.
- **Postamble** – A 1-bit value that is a don't care. This bit facilitates and Advanced to Standard Protocol change with some external tool implementations.

A minimum-length CP has a Preamble, a 2-bit Body, and a Postamble. This can be as simple as four logic zero values.

### 49.8.7 Scan Packet

A Scan Packet defines the period of TAP.7 pin activity where control information exchanged by the external tool and TAP.7 controller affects the TAPC state, except when the SP precedes a Change Packet. The external tool should view the Scan Packet as:

- Exchanging only the necessary TAPC information (TMS, TDI, and TDO or some subset thereof)
- Causing an advance of the TAP.7 TAPC state when it is not followed by a Change Packet.
- Causing an advance of the STL TAPC state when it is not followed by a Change Packet and when the STL is coupled.

With this perspective, the external tool views Scan Packets as running a virtual TAPC state machine at the TAP.7 pins. The effects of each Scan Packet take effect when the Scan Packet completes. The real TAP.7 and STL controller state machine states may change state before or after the virtual state machine.

### 49.8.8 SP Format

A SP serially exchanges the TDI, TMS, and TDO or a subset thereof, between the external tool and TAP.7 controller. Control information may be added to this exchange. A SP has three parts:

- **Payload** – Data + Control information (all scan formats), the payload content is varied to trade-off performance and flexibility. RDY bit(s) within the payload are included with some scan formats to provide the STL a means to stall progression of the TAPC state.
- **Delay** – Control information (optional) providing separation of adjacent payloads in time. Delay elements support simple external tool implementations where a delay element is used to stall progression of the TAPC state while it develops a new SP following the completion of the prior payload.

The header elements, delay elements, and control information within the payload element is consumed by the TAP.7 controller. This and other control information within the SP payload is not visible to the EPU and STL TAPCs.

### 49.8.8.1 Payload Element Content

#### 49.8.8.1.1 Input and Output Bit Frames

An input bit frame is sourced by the external tool and an output bit frame is sourced by the TAP.7 controller. The payload may contain an input bit frame, an output bit frame, or both. When an input bit frame arrives at the TAP.7 controller, the TDI and TMS information within it is presented in parallel to the TAP.1 controllers within the EPU and STL. The output bit frame begins when the input frame is completed. The payload is completed when the EPU and STL are ready to utilize this information and the TDO data needed to complete the output bit frame is available. This process is repeated for each SP.

With input bit frames, MScan and OScan scan formats have no control bits input bit frames.

With output bit frames:

- TDO information passed to the external tool is a logic 1 when the EPU indicates the TDO pin would be high impedance.
- RDY bits provide TAP.7 stall opportunities with MScan and some OScan formats.
- PC0 and PC1 bits are added to MScan scan formats to allow Wire-ANDed output.



### 49.8.8.1.2 Varying the content of input and output bit frames

The various scan formats (MScan, OScan0-7) provide a number of options for the serialization of scan information for use in a Star-2 scan topology. These scan formats vary the content of SPs to balance flexibility and performance for different use cases. The amount of information transferred per TAPC state is based on the constraints imposed by these use cases. The MScan and OScan scan formats determine the factors affecting the payload content as shown in the following table.

**Table 49-23. Factors determining the SP payload content**

| Scan format | Payload content determined by the: |        |
|-------------|------------------------------------|--------|
|             | TAPC state                         | Header |
| MScan       | –                                  | –      |
| OScan       | Yes                                | –      |

### 49.8.8.2 Delay Element Content

A delay element can be zero, one, two, or n bits in length. One and two bit delay elements are fixed length with neither the external tool or TAP.7 controller driving the TMS pin. The TMS pin is kept at the last value driven by either the external tool or TS. With an n-bit delay elements, the external tool drives the TMS pin for the first n-1 bits and may drive the TMS pin the nth bit. With an n-bit delay element the string of delay element bits is a series of directives virtually the same as those used by a CP (DLY\_NOP, DLY\_END, and DLY\_RES).

### 49.8.8.3 Scan Formats

#### 49.8.8.3.1 Overview

Scan formats utilizing the Advanced Protocol address different use cases. The amount of information transferred per TAPC state is based on the constraints imposed by these use cases. In one extreme two types of control information, the first allowing the stall of the completion of the Scan Packet, and the second supporting the Wire-AND of TAP.7 controller output are added to the TMS, TDI, and TDO information within the Scan Packet. In the other extreme, the external tool may minimize the information transferred for each TAPC state using its understanding of the application and transfer only the data of interest. In some cases the transfer of only one bit of information per TAPC is

required. There are a number of scan formats that operate between these two extremes. The external tool chooses a scan format that best matches the performance/functionality needs of the application.

The scan formats added to specify the use of the Advance Protocol are listed below:

- **MScan** – Maximum flexibility.
- **OScan0-7** – Optimized for test and debug applications.

These scan formats provide seven different functions. Five factors have influenced the definition of these scan formats:

- Supporting the capabilities of the IEEE1149.x and IEEE 1532 Standards.
- Providing the STL and external tool an opportunity to stall the TAPC state progression.
- Providing a minimum pin count and scalable functionality.
- Maximizing the efficiency of scan operations utilized for applications debug operations.
- Providing a rich set of capability for high end applications debug.

The performance and flexibility of MScan and OScan scan formats are shown in [Table 49-24](#) and [Table 49-26](#). The performance and flexibility of these scan formats are given subjective/approximate ratings in these tables. These ratings range from least to best. Use cases for these scan formats are also included in these tables.

#### 49.8.8.3.2 MScan scan formats

The MScan scan format provides a universal scan function. It is the most flexible but lowest-performance scan format. The Scan Packets used by these scan formats are packed with data and control information and is the same for all TAPC states. This scan format accommodates the deployment of IEEE 1149.1 IP with non-compliant behavior within the STL. The STL is allowed to stall the TAPC state progression (to pace the arrival of Scan Packets with its ability to process them), and uses Wire-ANDed Output.

**Table 49-24. MScan scan format use case summary**

| Scan format | Supporting                    | Performance | Flexibility | TCKC source         |
|-------------|-------------------------------|-------------|-------------|---------------------|
| MScan       | STL Stalls                    | Least       | Best        | External tool or TS |
|             | Test                          |             |             |                     |
|             | Non-compliant behavior STL IP |             |             |                     |

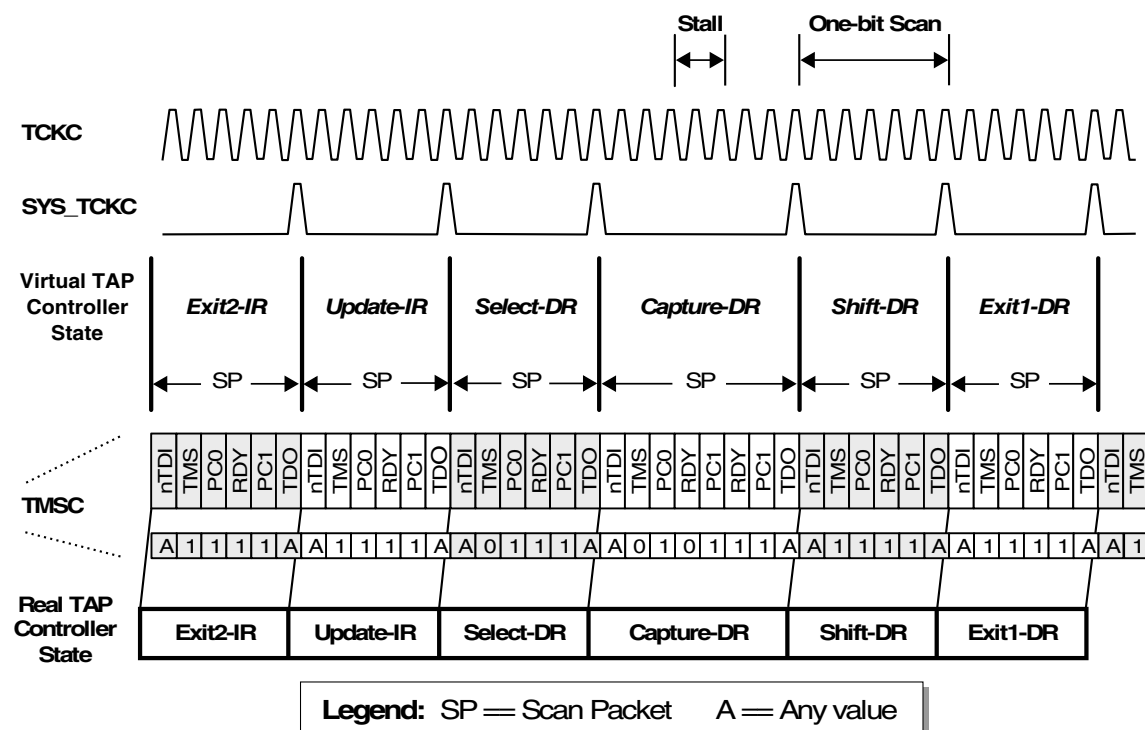
*Table continues on the next page...*

**Table 49-24. MScan scan format use case summary (continued)**

| Scan format | Supporting                     | Performance | Flexibility | TCKC source |
|-------------|--------------------------------|-------------|-------------|-------------|
|             | Multi-chip debug communication |             |             |             |
|             | Controller ID Allocation       |             |             |             |

### Table 49-25. MScan Scan Packet content

|       |                                 |     |     |     |     |     |  |                             |     |     |     |     |     |
|-------|---------------------------------|-----|-----|-----|-----|-----|--|-----------------------------|-----|-----|-----|-----|-----|
|       | Non-shift TAP controller states |     |     |     |     |     |  | Shift TAP controller states |     |     |     |     |     |
| MScan | nTDI                            | TMS | PC0 | RDY | PC1 | TDO |  | nTDI                        | TMS | PC0 | RDY | PC1 | TDO |



### Figure 49-18. MScan Transaction

### 49.8.8.3.3 OScan Scan Formats

The OScan scan formats provide optimization choices where no knowledge of the data content of the scan exchange is needed. The Scan Packet content of these scan formats is modulated by a combination of the scan format and the TAPC state for some OScan scan formats. For instance, TDI data and TDO data is not exchanged for non-shift TAPC states and included in shift TAPC states in certain OScan formats. The modulation of Scan Packet content with TAPC state varies by scan format.

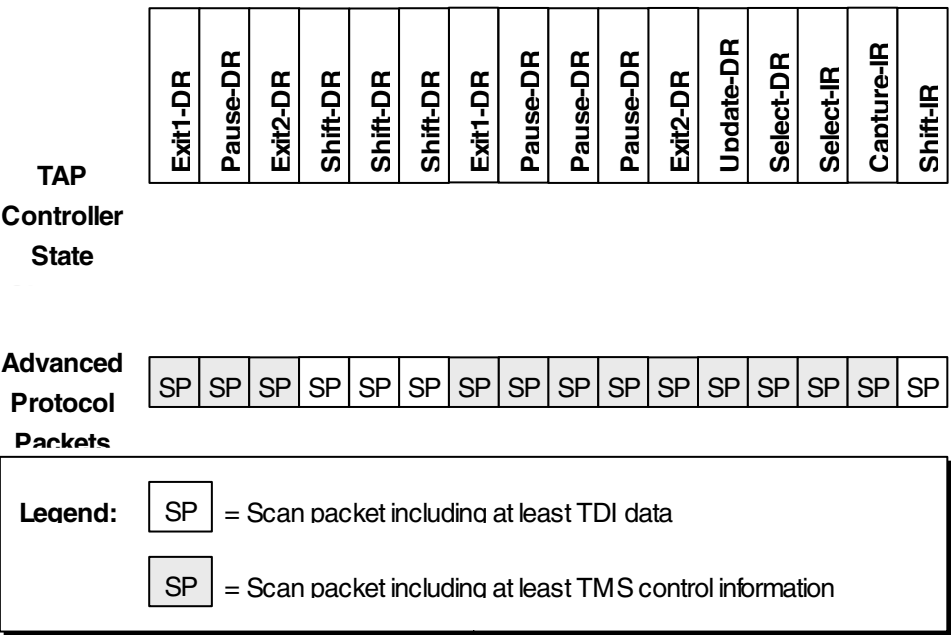


Figure 49-19. OScan Scan Packet content/TAPC state relationships

OScan scan formats replicate four functions in two forms:

- **OScan0 – OScan3:** Providing the best performance for a test clock source by the target system.
- **OScan 4 – OScan7:** Providing the best performance for a test clock sourced by the external tool.

The OScan4 – OScan7 scan formats are more efficient, as EOT escape sequences are used to identify an exit from the Shift-xR TAPC state. A Scan Packet associated with the Shift-xR TAPC state does not include a TMS bit. Instead, and EOT escape sequence overlaid on an nTDI bit causes an exit from shift TAPC state with no exit from this state occurring otherwise. A look at the 32-bit DR-Scan shows the impact of this optimization. A total of 96 bits are required when each Scan Packet associated with a Shift TAPC state when TMS is included in these Scan Packets while a total of 64 bits are required when TMS is not included in these Scan Packets. Assuming the EOT escape adds the equivalent of three bit periods, the number of TCK periods fo this transfer is 67/96 or roughly 70% of the total without this optimization.

These scan formats may only be used with an external tool sourced TCKC as EOTs are utilized to improve its performance. The OScan0 – OScan3 Scan Formats always have a TMS bit included in the Scan Packet. The OScan scan format use case summary is shown in the following table.

**Table 49-26. OScan scan format use case summary**

| Scan Format | Supporting                   | Performance  | Flexibility  | TCKC Source         |
|-------------|------------------------------|--------------|--------------|---------------------|
| OScan0      | Stalls                       | <div>↓</div> | Best         | External tool       |
| OScan1      | Test                         |              | <div>↑</div> |                     |
| OScan2      | Debug Performance            |              |              |                     |
| OScan3      | Debug Downloads (input only) | Best         | <div>↑</div> | External tool or TS |
| OScan4      | Stalls                       | <div>↓</div> |              |                     |
| OScan5      | Test                         |              |              |                     |
| OScan6      | Debug Performance            |              |              |                     |
| OScan7      | Debug Downloads (input only) | Best         |              |                     |

Certain OScan optimizations are applied in a hierarchical manner.

- **Optimization I:** OScan1/OScan5 – RDY bit(s) and their trailing ONE bits are deleted from all Scan Packets.
- **Optimization II:** OScan2/OScan6 – Optimization I + nTDI and TDO bits are deleted from Scan Packets associated with non-shift TAPC states.
- **Optimization III:** OScan3/OScan7 – Optimization II + TDO bits are deleted from Scan Packets.
- **Optimization IV:** OScan4 – OScan7 – TMS bits are deleted from Scan Packets associated with shift TAPC states.

**Table 49-27. OScan Scan Format "Payload"**

| Scan Format | Non-shift TAP controller states |     |     |     |        | Shift TAP controller states |     |     |     |
|-------------|---------------------------------|-----|-----|-----|--------|-----------------------------|-----|-----|-----|
| OScan0      | nTDI                            | TMS | RDY | TDO |        | nTDI                        | TMS | RDY | TDO |
| OScan1      | nTDI                            | TMS | –   | TDO |        | nTDI                        | TMS | –   | TDO |
| OScan2      | –                               | TMS | –   | –   |        | nTDI                        | TMS | –   | TDO |
| OScan3      | –                               | TMS | –   | –   |        | nTDI                        | TMS | –   | –   |
| OScan4      | nTDI                            | TMS | RDY | TDO |        | nTDI                        | –   | RDY | TDO |
| OScan5      | nTDI                            | TMS | –   | TDO |        | nTDI                        | –   | –   | TDO |
| OScan6      | –                               | TMS | –   | –   | Note 1 | nTDI                        | TMS | –   | TDO |
|             |                                 |     |     |     | Note 2 | nTDI                        | –   | –   | TDO |
| OScan7      | –                               | TMS | –   | –   |        | nTDI                        | –   | –   | –   |

Note 1: Following Capture-xR or Exit2-xR Scan Packet

Note 2: Following Shift-xR Scan Packet

## 49.9 Functional Description

This section combines the information given in the Ancillary Services, EPU Operation, and APU Operation sections into a single reference for performing common TAP.7 tasks.

### 49.9.1 Switching from Standard Protocol to Advanced Protocol

Upon exit of Type-0 to Type-4 reset, the CJTAG begins operation in Standard Protocol 4-pin mode. The following steps are required to move from Standard Protocol 4-pin mode to Advanced Protocol (either 4-pin or 2-pin) mode.

The first step required to move to the Advanced Protocol is to change the CJTAG control level to control level 2 via zero-bit scans (ZBS), as described in [Control levels](#).

After the control level is set to control level 2, TAP.7 commands can be executed as described in [EPU Commands](#). Execution of the Store Scan Format (STFMT) to set the scan format to any OSCAN0-7 or MSCAN format begins the switch from the Standard Protocol to the Advanced Protocol.

Once the STFMT command is executed selecting an Advanced Protocol scan format, the CJTAG transitions to the intermediate Control Protocol mode. The Control Protocol uses TMS inputs to control Configuration Change Packets as described in [Control Process Active \(CPA\) Function](#) and [Configuration Change Packets \(CP\)](#). Loading a change packet body value of 00 (CP\_END) results in the switch to the Advanced Protocol.

If Advanced Protocol 2-pin operation is desired, the APFC register should be written to a value of 10 or 11 prior to the execution of the STFMT command that selects an advanced scan format. The APFC register is written via the Store Conditional 2-bit (STC2) command described in [Store Conditional 2-bit \(STC2\)](#).

# Chapter 50

## JTAG Data Communication (JDC)

### 50.1 Introduction

The JTAG Data Communication (JDC) module provides the capability to move register data between the IPS and JTAG domains. This facilitates communication between internal resources that access memory mapped register space and an external tool that accesses the JTAG port.

### 50.2 Overview

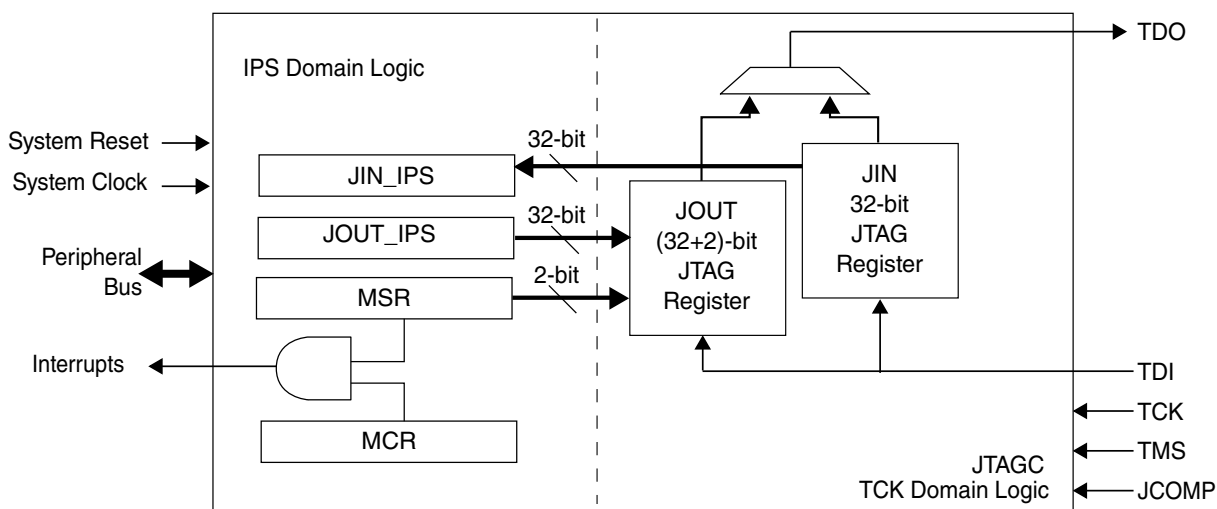
The JDC module consists of IPS accessible registers, JTAG accessible registers, and associated logic to coordinate movement of data from one register domain to another.

The JDC implements the following IPS data registers, occupying separate memory space:

- 32-bit memory mapped register that can be read or written via IPS (JOUT\_IPS), and whose contents are ported out for capture into a JTAG register (JOUT) to be read via the JTAG port.
- 32-bit memory mapped register that can only be read via IPS (JIN\_IPS), and whose contents are loaded from a JTAG register (JIN).

In addition to the data registers themselves, logic is implemented to indicate when new data has been shifted in via the JTAG port and is ready to be read from the JIN\_IPS register, and when new data has been written to the JOUT\_IPS register and is ready to be read via the JTAG port. The state of these flags is captured into a status register (MSR), and also provided to a JTAG register (JOUT) for tool visibility.

There is a single bus interface to port register data out to the JTAGC, and a single bus interface to port data in from the JTAGC. The architecture is shown in the following figure.



### Figure 50-1. JDC block diagram

## 50.3 Memory mapped registers

Four 32-bit registers are implemented. Only 32-bit accesses are valid. The effects of access that are not 32 bits are not defined.

## JDC memory map

| Address offset (hex) | Register name                            | Width (in bits) | Access | Reset value | Section/<br>page            |
|----------------------|------------------------------------------|-----------------|--------|-------------|-----------------------------|
| 0                    | Module Configuration Register (JDC_MCR)  | 32              | R/W    | 0000_0000h  | <a href="#">50.3.1/2325</a> |
| 4                    | Module Status Register (JDC_MSR)         | 32              | R/W    | 0000_0000h  | <a href="#">50.3.2/2326</a> |
| 8                    | JTAG Output Data Register (JDC_JOUT_IPS) | 32              | R/W    | 0000_0000h  | <a href="#">50.3.3/2327</a> |
| C                    | JTAG Input Data Register (JDC_JIN_IPS)   | 32              | R      | 0000_0000h  | <a href="#">50.3.4/2328</a> |



### 50.3.1 Module Configuration Register (JDC\_MCR)

The MCR is used to enable the interrupt outputs of the JDC. This register is reset by system functional reset.

Address: 0h base + 0h offset = 0h

|       |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |
|-------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|
| Bit   | 0        | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15       |
| R     | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    | JIN_IEN  |
| W     |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |
| Reset | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0        |
| Bit   | 16       | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31       |
| R     | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    |    | JOUT_IEN |
| W     |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |          |
| Reset | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0        |

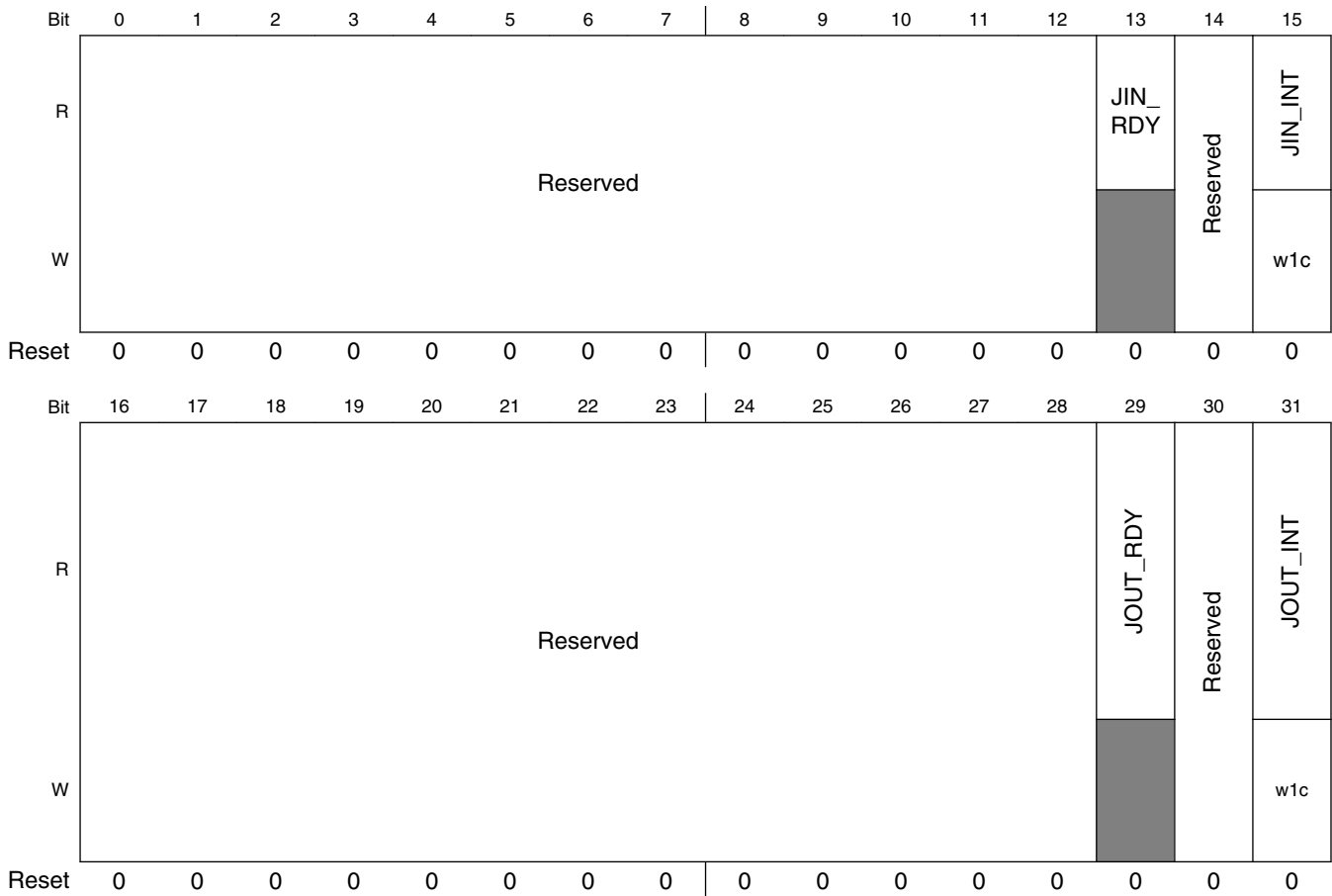
#### JDC\_MCR field descriptions

| Field             | Description                                                                                                                                              |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–14<br>Reserved  | This field is reserved.                                                                                                                                  |
| 15<br>JIN_IEN     | JIN Interrupt Enable.<br>0 Setting of MSR[JIN_INT] bit does not assert the JIN interrupt<br>1 Setting of MSR[JIN_INT] bit asserts the JIN interrupt      |
| 16–30<br>Reserved | This field is reserved.                                                                                                                                  |
| 31<br>JOUT_IEN    | JOUT Interrupt Enable.<br>0 Setting of MSR[JOUT_INT] bit does not assert the JOUT interrupt<br>1 Setting of MSR[JOUT_INT] bit asserts the JOUT interrupt |

50.3.2 Module Status Register (JDC\_MSR)

The MSR holds the JTAG register status and interrupt bits. This register is reset by system functional reset.

Address: 0h base + 4h offset = 4h



JDC\_MSR field descriptions

| Field            | Description                                                                                                                                  |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| 0–12<br>Reserved | This field is reserved.                                                                                                                      |
| 13<br>JIN_RDY    | JIN Ready (read only).<br>0 Cleared upon software read of JIN_IPS contents via IPS<br>1 Set when new data is written to the JIN_IPS register |
| 14<br>Reserved   | This field is reserved.                                                                                                                      |
| 15<br>JIN_INT    | JIN Interrupt.                                                                                                                               |

Table continues on the next page...

**JDC\_MSR field descriptions (continued)**

| Field             | Description                                                                                                                                   |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
|                   | 0 Cleared by writing logic 1<br>1 Set when new data is written to the JIN_IPS register                                                        |
| 16–28<br>Reserved | This field is reserved.                                                                                                                       |
| 29<br>JOUT_RDY    | JOUT Ready (read only).<br>0 Cleared upon tool read of JOUT register via JTAG port<br>1 Set when new data is written to the JOUT_IPS register |
| 30<br>Reserved    | This field is reserved.                                                                                                                       |
| 31<br>JOUT_INT    | JOUT Interrupt.<br>0 Cleared by writing logic 1<br>1 Set when JOUT_RDY bit is cleared by tool reading JOUT register                           |

**50.3.3 JTAG Output Data Register (JDC\_JOUT\_IPS)**

The JOUT\_IPS register holds data written via IPS. The JOUT\_IPS contents are ported out and captured into the JOUT register to be read via the JTAG port. This register is reset by system functional reset.

Address: 0h base + 8h offset = 8h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

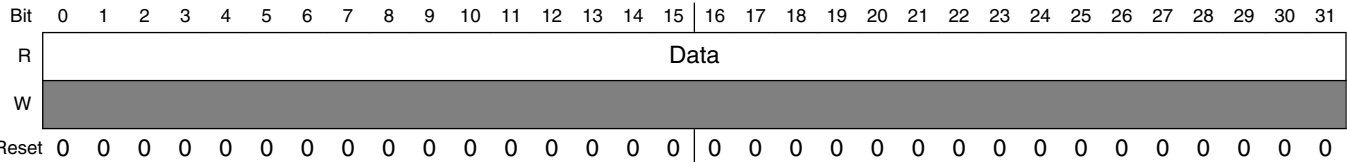
**JDC\_JOUT\_IPS field descriptions**

| Field        | Description    |
|--------------|----------------|
| 0–31<br>Data | JOUT_IPS data. |

50.3.4 JTAG Input Data Register (JDC\_JIN\_IPS)

Data written to the JTAG input data register (JIN) via the JTAG port is held in the JIN\_IPS register, where it can be read via IPS. This register is reset by system functional reset.

Address: 0h base + Ch offset = Ch



JDC\_JIN\_IPS field descriptions

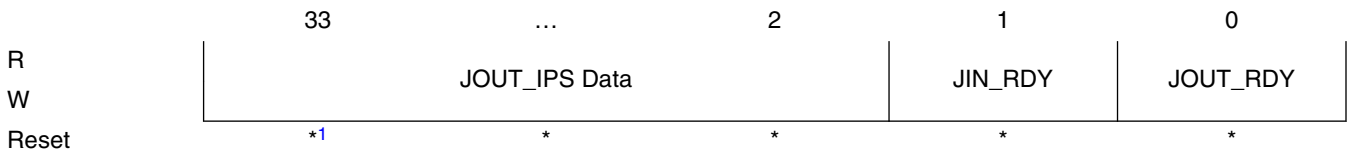
| Field        | Description   |
|--------------|---------------|
| 0–31<br>Data | JIN_IPS data. |

50.4 Non-Memory mapped register definition

The JDC also implements two JTAG accessible registers that are not memory mapped. One JTAG register is used to shift in data to be placed in the JIN\_IPS register. The other JTAG register captures data from the JOUT\_IPS register and ready status from the MSR to be shifted out via the JTAG port.

50.4.1 JTAG output data register (JOUT)

The JOUT register captures data from the JOUT\_IPS register upon execution of the JOUT\_READ JTAG instruction. It also holds the JIN\_RDY and JOUT\_RDY status bits. This register is reset by system power-on reset and JTAG reset. The following figure shows the format of the JOUT register.



1. The reset value of the JOUT register is 34'b0.

The JOUT register is described in the following table.

**Table 50-6. JOUT JTAG register field descriptions**

| Field         | Description                       |
|---------------|-----------------------------------|
| JOUT_IPS Data | Data value from JOUT_IPS register |
| JIN_RDY       | State of JIN_RDY bit from MSR     |
| JOUT_RDY      | State of JOUT_RDY bit from MSR    |

## 50.4.2 JTAG input data register (JIN)

Data is written to the JIN register via JTAG during execution of the JIN\_WRITE JTAG instruction. The JIN data is later captured in the JIN\_IPS register to be read via IPS. This register is reset by system power-on reset and JTAG reset. The following figure shows the format of the JIN register.



1. The reset value of the JIN register is 32'b0.

The JIN register is described in the following table.

**Table 50-7. JIN JTAG register field descriptions**

| Field    | Description                                                                                                              |
|----------|--------------------------------------------------------------------------------------------------------------------------|
| JIN Data | Contains data to be captured in JIN_IPS register upon exit of Update-DR state when executing WRITE_JIN JTAG instruction. |

## 50.5 Functional description

The JDC module provides the ability to shift in data via the JTAG port and capture that data into a memory mapped register that can be accessed via IPS. It also provides the ability to capture data written to a memory mapped register into a JTAG shift register for output via the JTAG port. An overview of the module functionality is described below.

- A software write to the JOUT\_IPS register sets the JOUT\_RDY flag bit, indicating new data is available to be read from the JOUT register via the JTAG port. The state of the JOUT\_RDY bit is reflected in the MSR and also ported out to the JOUT register. A JTAG read of the JOUT register via execution of the JOUT\_READ

instruction with a JOUT\_RDY bit whose value is logic 1 indicates the register contains new data. The JOUT\_RDY flag bit is cleared upon exit of the Capture-DR JTAG state during execution of the JOUT\_READ instruction. Clearing the JOUT\_RDY bit indicates to software that a new data value can be written to the JOUT\_IPS register.

- A JTAG write to the JIN register via execution of the JIN\_READ JTAG instruction updates the contents of the JIN\_IPS register upon exit of the Update-DR state. An update of the JIN\_IPS register sets the JIN\_RDY flag bit, indicating new data is available to be read via IPS. The state of the JIN\_RDY bit is reflected in the MSR register and also ported out to the JOUT register. The JIN\_RDY flag bit is cleared upon software read of the JIN\_IPS register. A JTAG read of the JOUT register with a JIN\_RDY value of logic 0 indicates new data can be written to the JIN register.

## 50.5.1 JTAG register access

Refer to the JTAGC documentation for information on how to access the JTAG registers.

### 50.5.1.1 JDC block instructions

The JDC block implements the IEEE 1149.1-2001 defined instructions listed in the following table. This section gives an overview of each instruction; refer to the IEEE 1149.1-2001 standard for more details. All undefined opcodes are reserved.

**Table 50-8. JTAG instructions**

| Instruction | Code[4:0]         | Instruction Summary                                                                                                                          |
|-------------|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| Reserved    | 00001             | Factory debug reserved                                                                                                                       |
| JOUT_READ   | 00010             | Selects JOUT data register. Data from JOUT_IPS is captured into JOUT data register upon entry to Capture-DR state while JOUT_READ is active. |
| JIN_WRITE   | 01110             | Selects JIN data register. Data from JIN is captured into JIN_IPS upon exit of Update-DR state while JIN_WRITE is active.                    |
| BYPASS      | 11111             | Selects bypass register for data operations                                                                                                  |
| Reserved    | All other opcodes | Decoded to select bypass register                                                                                                            |

# Chapter 51

## Nexus Development Interface (NDI)

### 51.1 Overview

The NDI integrates seven individual Nexus modules that provide the chip's development support interface.

All seven Nexus modules are connected to the Nexus Port Controller (NPC). The following table lists the modules.

**Table 51-1. Nexus modules of NDI**

| Nexus modules                                                                                                         | For detailed information, see                                                               |
|-----------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|
| Two Nexus e200z7 core interfaces (NZ7C3): one each for Core0 and Core1                                                | <i>e200z759n3 Core Reference Manual</i> (document number e200z759n3CRM)                     |
| Two Nexus eTPU development interfaces (NDEDI and NSEDI)                                                               | <a href="#">Nexus eTPU Development Interface (NDEDI)</a>                                    |
| Two Nexus crossbar eDMA interfaces (NXDM): one each for eDMA_A and eDMA_B<br>One Nexus crossbar SIPI interface (NXSI) | <a href="#">Nexus eDMA Interface (NXDM)</a> and <a href="#">Nexus SIPI Interface (NXSI)</a> |

The chip has no debug capability for the CSE module. CSE is not traceable.

#### NOTE

In each core, simultaneous access to debug control registers by software and Nexus hardware is not possible. When debug registers are allocated to Nexus hardware, they cannot be accessed by software.

## 51.2 Nexus clients for TAP access

After the NDI is out of the reset state: To grant the NDI ownership of the TAP, a specific instruction must be loaded in the JTAG Controller (JTAGC) module. Each Nexus client has its own JTAGC instruction opcode for ownership of the TAP, granting that client read/write access to the TAP's registers. The following table shows the JTAGC instruction opcode for each Nexus client. After the JTAGC opcode for a client has been loaded, the client is enabled by loading its NEXUS-ENABLE instruction.

**Table 51-2. JTAG client select instructions**

| JTAGC instruction         | Opcode | Description                                                                  |
|---------------------------|--------|------------------------------------------------------------------------------|
| ACCESS_AUX_NPC            | 10000  | Enables access to the NPC TAP controller                                     |
| ACCESS_AUX_ONCE           | 10001  | Enables access to the primary OnCE TAP controller (Core0)                    |
| ACCESS_AUX_eTPU           | 10010  | Enables access to the eTPU Nexus TAP controller (ETPU A/B)                   |
| ACCESS_AUX_NXDM           | 10011  | Enables access to the eDMA Nexus TAP controller (eDMA_A)                     |
| Reserved                  | 10100  | Not used                                                                     |
| ACCESS_AUX_SIPI           | 10101  | Enables access to the SIPI Nexus TAP controller                              |
| ACCESS_AUX_eTPU_Secondary | 10110  | Enables access to secondary ETPU Nexus TAP controller (ETPU C)               |
| ACCESS_AUX_NXDM_B         | 10111  | Enables access to the second eDMA Nexus TAP controller (eDMA_B)              |
| Reserved                  | 11000  | Not used                                                                     |
| ACCESS_AUX_ONCE_1         | 11001  | Enables access to the secondary OnCE TAP controller (Core1)                  |
| ACCESS_AUX_PAR            | 11010  | Enables parallel access to both OnCE TAP controllers (Core0 and Core1)       |
| ACCESS_AUX_MULTI          | 11100  | Enables serial access to both OnCE TAP controllers (Core0 followed by Core1) |
| ACCESS_AUX_JDC            | 11101  | Enables access to the JDC TAP controller                                     |
| Reserved                  | 11110  | Not used                                                                     |
| BYPASS                    | 11111  | Selects bypass register for data operations                                  |

## 51.3 Debug break and cross-triggering

This chip supports central break control among the cores and debug modules such as eTPU, eDMA, and Watchdog, allowing synchronous restart control for all cores when exiting debug mode.



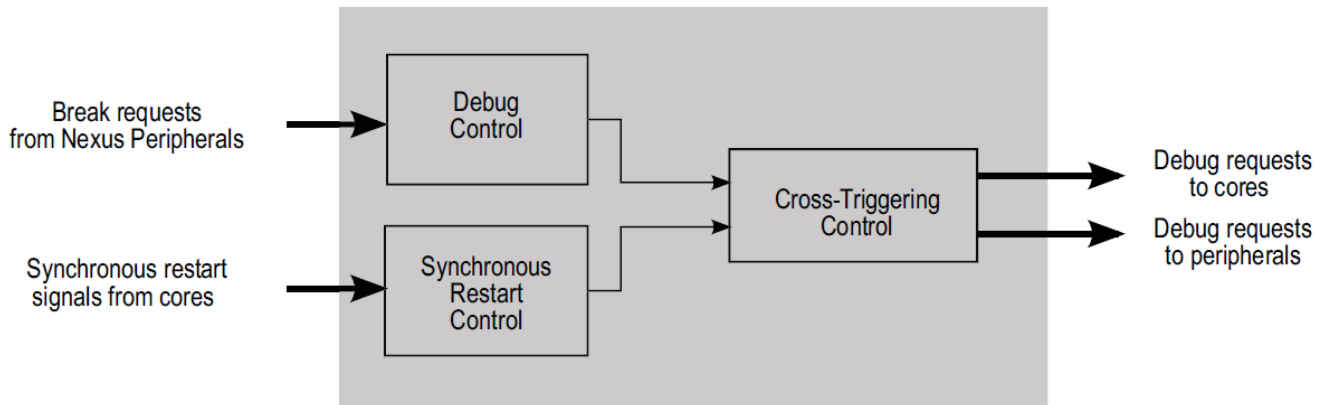


**Figure 51-1. Debug break and cross-triggering logic**

### 51.3.1 Break control

Multiple modules on the chip, including the Nexus trace blocks, have debug features that allow them to request device-wide breaks to enter debug mode. All debug mode entry requests are combined (logically ORed) together, and the result produces the debug requests to the cores and peripherals.

The cross-triggering logic generates these debug request signals:

- Debug Request signal (active high) for all non-core peripherals
- External Debug Request (active high) for requesting an external debug event to cores

### 51.3.2 Cross-triggering control

Cross-triggering provides the ability to select whether modules enter debug mode upon receiving halt request signals from Nexus modules. In response to debug mode entry, some modules can be programmed to halt operation or to continue normal operation. This cross-triggering feature is implemented through debug control bits in the modules. The cores and peripherals can be programmed selectively with their respective control bits.

### 51.3.3 Synchronous restart control

The cross-triggering logic implements a feature that enables all debug peripherals—cores and peripherals that are put in the debug state—to restart synchronously, as follows:

1. The debugging tool sequentially scans the CPUSCR for each core to restore the proper processor state.
2. The debugging tool scans into the main chip TAP controller the instruction whose execution selects all core TAP controllers in parallel.
3. To enable any combination of cores to exit debug mode synchronously: The debugging tool simultaneously scans into each core's TAP instruction register (OCMD) the 10-bit sequence 180h, which sets both the GO and EX fields to 1 and clears RS[0:6] field to 0.
4. If a core exits debug mode, it sets a flag OnCE go+exit command.
5. When the OR of the core OnCE go+exit command signals is asserted, all other selected non-cores and peripherals in debug synchronously exit debug mode.

## **51.4 Nexus eDMA Interface (NXDM) and Nexus SIPI Interface (NXSI)**

The e200z7 eDMA Nexus module (NXDM) and SIPI module (NXSI) comply with the Class 3 defined data trace feature of the IEEE-ISTO 5001-2003 standard.

### **51.4.1 Introduction**

The NXDM can be programmed to trace data accesses for the eDMA module on the system bus. The NXSI can be programmed to trace data accesses for the SIPI module on the system bus. This eDMA module and SIPI module as well as the Nexus module are components of the e200z7 platform. All output messages and register accesses are compliant with the protocol defined in the IEEE-ISTO 5001 standard.

#### **NOTE**

The auxiliary port and its signals, such as MCKO, MSEO[1:0], MDO[15:0] and others, are referenced. The device NPC module arbitrates the access of the single auxiliary port. The functions of the NXDM and SIPI modules are described without the interaction of the NPC, as if Nexus has a dedicated auxiliary port, to simplify the description.

#### **51.4.1.1 Block Diagrams**

Figure 51-2 shows a block diagram of the NXDM. The block diagram of the NXSI is the same.

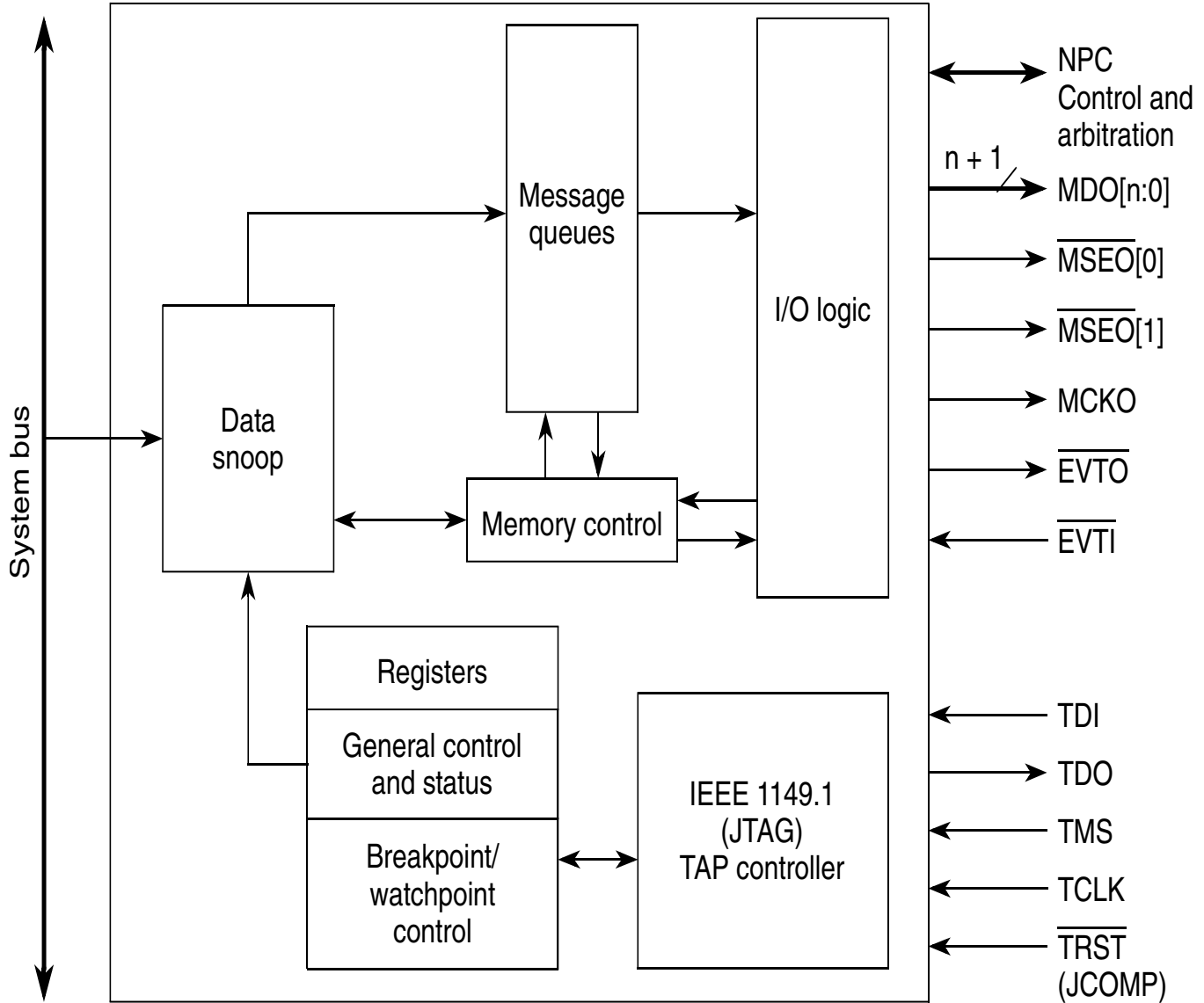


Figure 51-2. NXDM and NXSI Block Diagram

### 51.4.1.2 Features

Features include the following:

- Data trace via data write messaging (DWM) and data read messaging (DRM). This provides the capability for the development tool to trace reads and/or writes through the eDMA and SIPI modules to (selected) internal memory resources.
- Watchpoint messaging via the auxiliary pins.
- Watchpoint trigger enable of data trace messaging (DTM).

- Registers for data trace, watchpoint generation, and watchpoint trigger.
- All features controllable and configurable via the JTAG port.
- Power management.
  - Low power design
  - Dynamic power management of FIFOs and control logic

### 51.4.2 External Signal Description

The NXDM and NXSI modules use the same pins and pin protocol as the JTAG Controller and NPC modules.

#### 51.4.2.1 Rules for Output Messages

The NXDM and NXSI modules observe the same rules for output messages as the NPC.

#### 51.4.2.2 Auxiliary Port Arbitration

The NXDM and NXSI modules arbitrate for the shared Nexus port. This arbitration is handled by the NPC based on prioritized requests from the NXDM, NXSI, and the other Nexus clients sharing the port.

### 51.4.3 NXDM and NXSI Programmers Model

This section describes the programmers model. Nexus registers are accessed using the JTAG port in compliance with IEEE 1149.1. See the JTAGC module details for more information about Nexus register access.

#### 51.4.3.1 NXDM and NXSI Nexus Register Map

Table 51-3. NXDM and NXSI Register Map

| Nexus Register                           | Nexus Access Opcode | Read/Write | Read Address | Write Address |
|------------------------------------------|---------------------|------------|--------------|---------------|
| Client Select Control (CSC) <sup>1</sup> | 0x1                 | R          | 0x02         | -             |

Table continues on the next page...

**Table 51-3. NXDM and NXSI Register Map (continued)**

| Nexus Register                                    | Nexus Access Opcode | Read/Write | Read Address | Write Address |
|---------------------------------------------------|---------------------|------------|--------------|---------------|
| Port Configuration Register (PCR) <sup>1</sup>    | Refer to NPC        | R/W        | -            | -             |
| Development Control 1 (DC1_n)                     | 0x2                 | R/W        | 0x04         | 0x05          |
| Development Control 2 (DC2_n)                     | 0x3                 | R/W        | 0x05         | 0x06          |
| Watchpoint Trigger (WT_n)                         | 0xB                 | R/W        | 0x16         | 0x17          |
| Data Trace Control (DTC_n)                        | 0xD                 | R/W        | 0x1A         | 0x1B          |
| Data Trace Start Address 1 (DTSA1_n)              | 0xE                 | R/W        | 0x1C         | 0x1D          |
| Data Trace Start Address 2 (DTSA2_n)              | 0xF                 | R/W        | 0x1E         | 0x1F          |
| Data Trace End Address 1 (DTEA1_n)                | 0x12                | R/W        | 0x24         | 0x25          |
| Data Trace End Address 2 (DTEA2_n)                | 0x13                | R/W        | 0x26         | 0x27          |
| Breakpoint/Watchpoint Control Register 1 (BWC1_n) | 0x16                | R/W        | 0x2C         | 0x2D          |
| Breakpoint/Watchpoint Control Register 2 (BWC2_n) | 0x17                | R/W        | 0x2E         | 0x2F          |
| Breakpoint/Watchpoint Address Register 1 (BWA1_n) | 0x1E                | R/W        | 0x3C         | 0x3D          |
| Breakpoint/Watchpoint Address Register 2 (BWA2_n) | 0x1F                | R/W        | 0x3E         | 0x3F          |
| Reserved                                          | 0x20-0x3F           | -          | 0x40-0x7E    | 0x41-0x7F     |

1. The CSC and PCR registers are shown in this table as part of the Nexus programmer's model. They are only present at the top level Nexus3 controller (NPC), not in the NXDM or NXSI module. The device's CSC register is readable through Nexus3; the PCR is shown for reference only.

### 51.4.3.2 NXDM and NXSI Registers

Detailed register definitions for the NXDM and NXSI implementation are as follows:

#### 51.4.3.2.1 Development Control Register DC1

The development control registers control the basic development features of the NXDM and NXSI modules.

|       |     |         |    |    |    |    |    |     |    |    |    |    |    |    |    |    |
|-------|-----|---------|----|----|----|----|----|-----|----|----|----|----|----|----|----|----|
|       | 31  | 30      | 29 | 28 | 27 | 26 | 25 | 24  | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| R     | OPC | MCK_DIV |    |    |    | 0  | 0  | WEN | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| W     |     |         |    |    |    |    |    |     |    |    |    |    |    |    |    |    |
| Reset | 0   | 0       | 0  | 0  | 0  | 0  | 0  | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |    |    |    |    |    |   |   |   |   |   |     |   |   |    |   |
|-------|----|----|----|----|----|----|---|---|---|---|---|-----|---|---|----|---|
|       | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4   | 3 | 2 | 1  | 0 |
| R     | 0  | C  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 |     |   |   |    |   |
| W     |    |    |    |    |    |    |   |   |   |   |   | EIC |   |   | TM |   |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0   | 0 | 0 | 0  | 0 |

**Figure 51-3. Development Control Register 1 (DC1)**

**Table 51-4. DC1 Field Description**

| Field                      | Description                                                                                                                                                                                                                    |
|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31<br>OPC <sup>1</sup>     | Output port mode control.<br>0 Reduced port mode configuration<br>1 Full port mode configuration                                                                                                                               |
| 30-29 MCK_DIV <sup>1</sup> | Nexus message clock divide ratio.<br>00 MCKO is 1x system bus clock frequency<br>01 MCKO is 1/2x system bus clock frequency<br>10 MCKO is 1/4x system bus clock frequency<br>11 MCKO is 1/8x system bus clock frequency        |
| 28-27<br>EOC               | EVT $\overline{O}$ control.<br>00 $\overline{EVT\overline{O}}$ upon occurrence of watchpoint (internal or external)<br>01 $\overline{EVT\overline{O}}$ upon entry into system-level debug mode (ipg_debug)<br>1X Invalid value |
| 26-25                      | Reserved, read as 0.                                                                                                                                                                                                           |
| 24 WEN                     | Watchpoint trace enable.<br>0 Watchpoint messaging disabled<br>1 Watchpoint messaging enabled                                                                                                                                  |
| 23-5                       | Reserved, read as 0.                                                                                                                                                                                                           |
| 4-3 EIC                    | EVT $\overline{I}$ control.<br>00 $\overline{EVT\overline{I}}$ for synchronization (Data Trace)<br>01 Invalid value<br>10 $\overline{EVT\overline{I}}$ disabled for this module<br>11 Invalid value                            |
| 2-0 TM                     | Trace mode.<br>000 No Trace<br>1XX Program trace enabled (not supported) <sup>2</sup><br>X1X Data trace enabled<br>XX1 Ownership Trace enable (not supported) <sup>2</sup>                                                     |

1. The output port mode control bit (OPC) and MCKO divide bits (MCK\_DIV) are shown for clarity. These functions are controlled globally by the NPC port control register (PCR).
2. The XBAR bus Nexus trace module only supports data trace and watchpoint trace, and does not support program or ownership trace.

### 51.4.3.2.2 Development Control Register 2 (DC2)

|       |  |             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|-------|--|-------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
|       |  | Access: R/W |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|       |  | 31          | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| R     |  | EWC         |    |    |    |    |    |    |    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| W     |  |             |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Reset |  | 0           | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 51-4. Development Control Register 2 (DC2)

Table 51-5. DC2 Field Description

| Field                     | Description                                                                                                                                                                                                                                                                                                                                                               |
|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31-24<br>EWC <sup>1</sup> | EVTO Watchpoint Configuration<br>00000000 = No watchpoints trigger $\overline{\text{EVTO}}$<br>1XXXXXXX = Invalid value<br>X1XXXXXX = Invalid value<br>XX1XXXXX = Invalid value<br>XXX1XXXX = Invalid value<br>XXXX1XXX = Internal watchpoint #1 triggers EVTO<br>XXXXX1XX = Internal watchpoint #2 triggers EVTO<br>XXXXXX1X = Invalid value<br>XXXXXXX1 = Invalid value |
| 23-0                      | Reserved, read as 0.                                                                                                                                                                                                                                                                                                                                                      |

1. The EOC bits in DC1 must be programmed to trigger  $\overline{\text{EVTO}}$  on watchpoint occurrence for the EWC bits to have any effect.

### 51.4.3.2.3 Watchpoint Trigger Register (WT)

The watchpoint trigger register allows the watchpoints defined internally to the NXDM and NXSI modules to trigger actions. These watchpoints can control data trace enable and disable. The WT bits can be used to produce an address related window for triggering trace messages.

|       | 31 | 30 | 29 | 28 | 27 | 26 | 25  | 24 | 23 | 22 | 21  | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|----|----|----|----|----|----|-----|----|----|----|-----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| R     | 0  | 0  | 0  | 0  | 0  | 0  | DTS |    |    |    | DTE |    |    |    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |   |
| W     |    |    |    |    |    |    |     |    |    |    |     |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0  | 0  | 0  | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Figure 51-5. Watchpoint Trigger Register (WT)

Table 51-6. WT Field Description

| Field        | Description                                                                                                                                                                                              |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31-26        | Reserved, read as 0.                                                                                                                                                                                     |
| 25-23<br>DTS | DTS - Data trace start control<br>000 Trigger disabled<br>001-100 Invalid value<br>101 Use internal watchpoint #1 (BWA1 register)<br>110 Use internal watchpoint #2 (BWA2 register)<br>111 Invalid value |
| 22-20<br>DTE | DTE - Data trace end control<br>000 Trigger disabled<br>001-100 Invalid value<br>101 Use internal watchpoint #1 (BWA1 register)<br>110 Use internal watchpoint #2 (BWA2 register)<br>111 Invalid value   |
| 19-0         | Reserved, read as 0.                                                                                                                                                                                     |

**NOTE**

The WT bits ONLY enable data trace if the tm bits within the development control register (DC) have not already been set to enable data trace.

**51.4.3.2.4 Data Trace Control Register (DTC)**

The data trace control register controls whether DTM Messages are restricted to reads, writes or both for a user programmable address range. There are two data trace channels controlled by the DTC for the NXDM and NXSI modules.



Access: R/W

|       | 31   | 30 | 29   | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7   | 6 | 5   | 4 | 3 | 2 | 1 | 0 |
|-------|------|----|------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|-----|---|-----|---|---|---|---|---|
| R     | RWT1 |    | RWT2 |    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | RC1 |   | RC2 |   | 0 | 0 | 0 | 0 |
| W     |      |    |      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |     |   |     |   |   |   |   |   |
| Reset | 0    | 0  | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0   | 0 | 0   | 0 | 0 | 0 | 0 | 0 |

**Figure 51-6. Data Trace Control Register (DTC)****Table 51-7. DTC Field Description**

| Bit           | Description                                                                                                                                               |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31-30<br>RWT1 | Read/write trace 1<br>00 No trace messages generated<br>X1 Enable data read trace<br>1X Enable data write trace                                           |
| 29-28<br>RWT2 | Read/write trace 2<br>00 No trace messages generated<br>X1 Enable data read trace<br>1X Enable data write trace                                           |
| 27-8          | Reserved, read as 0.                                                                                                                                      |
| 7<br>RC1      | Range control 1<br>0 Condition trace on address within range (endpoints inclusive)<br>1 Condition trace on address outside of range (endpoints exclusive) |
| 6<br>RC2      | Range control 2<br>0 Condition trace on address within range (endpoints inclusive)<br>1 Condition trace on address outside of range (endpoints exclusive) |
| 5-0           | Reserved, read as 0.                                                                                                                                      |

### 51.4.3.2.5 Data Trace Start Address Registers 1 and 2 (DTSA1 and DTSA2)

The data trace start address registers define the start addresses for each trace channel.

Access: R/W

|       | 31                       | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-------|--------------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| R     | DATA TRACE START ADDRESS |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| W     |                          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Reset | 0                        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 51-7. Data Trace Start Address Registers (DTSA1, DTSA2)**

### 51.4.3.2.6 Data Trace End Address Registers 1 and 2 (DTEA1 and DTEA2)

The data trace end address registers define the end addresses for each trace channel.

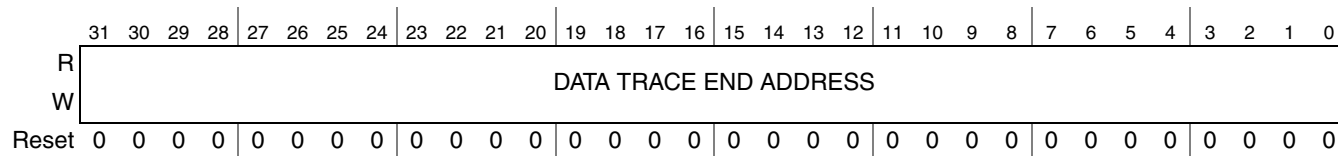
**Figure 51-8. Data Trace Start Address Registers (DTEA1, DTEA2)**

Table 51-8 illustrates the range that is selected for data trace for various cases of DTSA being less than, greater than, or equal to DTEA.

**Table 51-8. Data Trace Address Range Options**

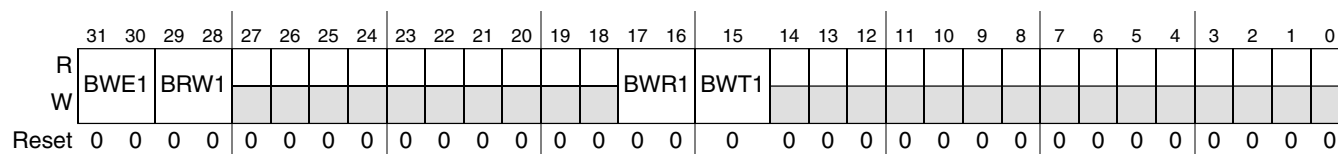
| Programmed Values | Range Control Bit Value | Range Selected          |
|-------------------|-------------------------|-------------------------|
| DTSA < or = DTEA  | 0                       | DTSA → ← DTEA           |
| DTSA < or = DTEA  | 1                       | ← DTSA DTEA →           |
| DTSA > DTEA       | -                       | Invalid range, no trace |

**NOTE**

DTSA must be less than (or equal to) DTEA to guarantee correct data write/read traces. When the range control bit is 0 (internal range), accesses to DTSA and DTEA addresses are traced. When the range control bit is 1 (external range), accesses to DTSA and DTEA are not traced.

**51.4.3.2.7 Breakpoint / Watchpoint Control Register 1 (BWC1)**

Breakpoint/watchpoint control register 1 controls attributes for generation of NXDM and NXSI watchpoint number 1.

**Figure 51-9. Break / Watchpoint Control Register 1 (BWC1)****Table 51-9. BWC1 Field Description**

| Field | Description                                                     |
|-------|-----------------------------------------------------------------|
| 31-30 | Breakpoint/watchpoint #1 enable                                 |
| BWE1  | 00 Internal Nexus watchpoint #1 disabled<br>01-10 Invalid value |

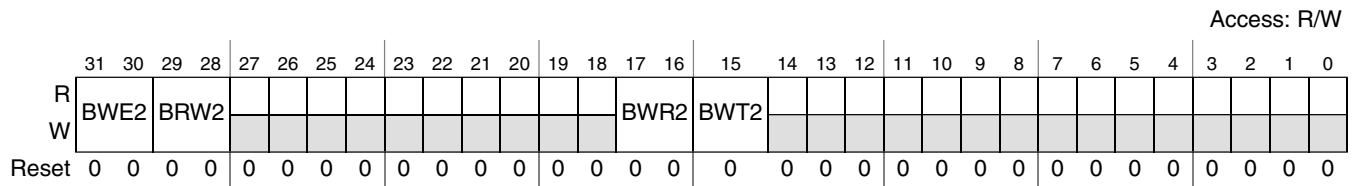
Table continues on the next page...

**Table 51-9. BWC1 Field Description (continued)**

| Field         | Description                                                                                                                                                                                     |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               | 11 Internal Nexus watchpoint #1 enabled                                                                                                                                                         |
| 29-28<br>BRW1 | Breakpoint/watchpoint #1 read/write select<br>00 Watchpoint #1 hit on read accesses<br>01 Watchpoint #1 hit on write accesses<br>10 Watchpoint #1 on read or write accesses<br>11 Invalid value |
| 27-18         | Reserved, read as 0.                                                                                                                                                                            |
| 17-16<br>BWR1 | Breakpoint/watchpoint #1 register compare<br>00 No register compare (same as BWC1[31:30] = 2'b00)<br>01 Invalid value<br>10 Compare with BWA1 value<br>11 Invalid value                         |
| 15<br>BWT1    | Breakpoint/watchpoint #1 type<br>0 Invalid value<br>1 Watchpoint #1 on data accesses                                                                                                            |
| 14-0          | Reserved, read as 0.                                                                                                                                                                            |

### 51.4.3.2.8 Breakpoint / Watchpoint Control Register 2 (BWC2)

Breakpoint/watchpoint control register2 controls attributes for generation of NXDM and NXSI watchpoint number 2.

**Figure 51-10. Break / Watchpoint Control Register 2 (BWC2)****Table 51-10. BWC1 Field Description**

| Field         | Description                                                                                                                                   |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| 31-30<br>BWE2 | Breakpoint/watchpoint #2 enable<br>00 Internal Nexus watchpoint #2 disabled<br>01-10 Invalid value<br>11 Internal Nexus watchpoint #2 enabled |
| 29-28<br>BRW2 | Breakpoint/watchpoint #1 read/write select<br>00 Watchpoint #2 hit on read accesses                                                           |

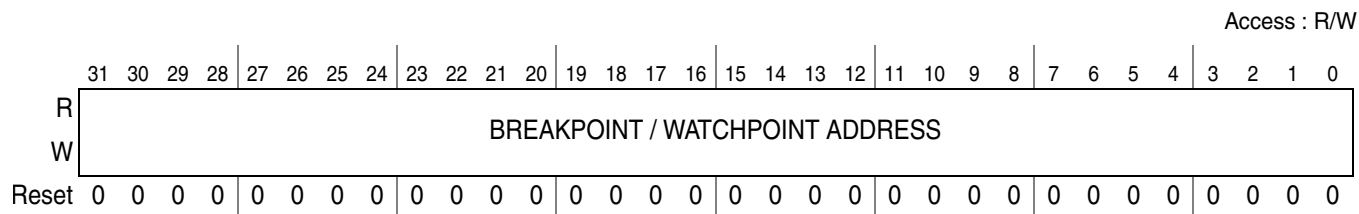
*Table continues on the next page...*

**Table 51-10. BWC1 Field Description (continued)**

| Field         | Description                                                                                                                                                             |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               | 01 Watchpoint #2 hit on write accesses<br>10 Watchpoint #2 on read or write accesses<br>11 Invalid value                                                                |
| 27-18         | Reserved, read as 0.                                                                                                                                                    |
| 17-16<br>BWR2 | Breakpoint/watchpoint #2 register compare<br>00 No register compare (same as BWC1[31:30] = 2'b00)<br>01 Invalid value<br>10 Compare with BWA2 value<br>11 Invalid value |
| 15<br>BWT2    | Breakpoint/watchpoint #2 type<br>0 Invalid value<br>1 Watchpoint #2 on data accesses                                                                                    |
| 14-0          | Reserved, read as 0.                                                                                                                                                    |

#### 51.4.3.2.9 Breakpoint/Watchpoint Address Registers 1 and 2 (BWA1 and BWA2)

The breakpoint/watchpoint address registers are compared with bus addresses to generate internal watchpoints.

**Figure 51-11. Breakpoint / Watchpoint Address Registers (BWA1, BWA2)**

#### 51.4.3.2.10 Unimplemented Registers

Unimplemented registers are those with client select and index value combinations other than the ones listed in [Table 51-3](#). For unimplemented registers, the NXDM and NXSI modules drive TDO to zero during the "SHIFT-DR" state. The modules also transmit an error message with the invalid access opcode encoding.

#### 51.4.3.2.11 Programming Considerations ( $\overline{\text{RESET}}$ )

If Nexus3 register configuration is to occur during system reset (as opposed to debug mode), all NXDM configuration should be completed between the negation of JCOMP and system reset de-assertion, after the JTAG DID register has been read by the tool.

### 51.4.3.2.12 IEEE 1149.1 (JTAG) Test Access Port

The NXDM and NXSI modules use the IEEE 1149.1 TAP controller for accessing Nexus resources. The JTAG signals themselves are shared by all TAP controllers on the device. See the JTAGC module details for more information about the JTAG interface.

The NXDM and NXSI modules implement a 4-bit instruction register (IR). The valid instructions and method for register access are outlined in the JTAGC module details.

#### 51.4.3.2.12.1 NXDM and NXSI JTAG DID Register

This JTAG DID register that is included in the NXDM and NXSI modules provides key development attributes to the development tool concerning the NXDM and NXSI blocks. The register is accessed through the standard JTAG IR/DR paths. Refer to the PMC chapter.

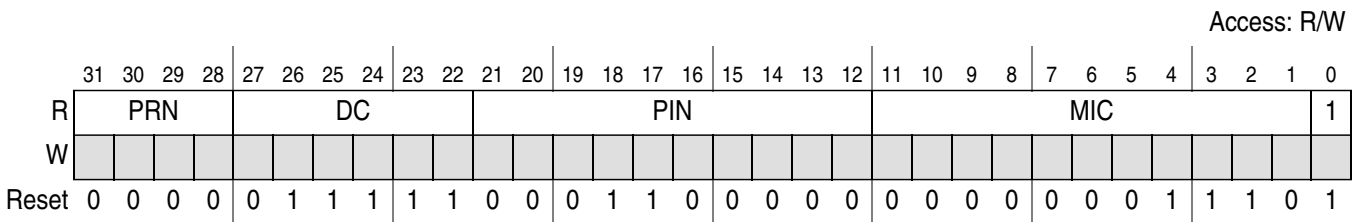


**Figure 51-12. NXDM and NXSI JTAG DID Register**

**Table 51-11. NXDM and NXSI JTAG DID Field Descriptions**

| Field                     | Description                                                                       |
|---------------------------|-----------------------------------------------------------------------------------|
| 31-28<br>PRN <sup>1</sup> | Embedded part revision number (0x0)                                               |
| 27-22<br>DC               | Freescale design center ID number (0x1F)                                          |
| 21-12<br>PIN              | NXDM and NXSI module part identification number, defines the features set. (0x60) |
| 11-1<br>MIC               | Manufacturer identity code<br>0x00E Freescale                                     |
| 0                         | Fixed per JTAG 1149.1<br>1 Always set                                             |

1. The revision number is initially 0 and could change in the future.

51.4.3.2.12.2 Enabling the NXDM and NXSI TAP Controllers

Assertion of a power-on-reset signal or assertion of the JCOMP pin resets all TAP controllers on the device. Upon exit from the test-logic-reset state, the IR value is loaded with the JTAG DID. When the NXDM or NXSI TAP is accessed, this information helps the development tool obtain information about the Nexus module it is accessing, such as version, sequence, feature set, and so forth.

51.4.3.2.12.3 NXDM and NXSI Register Access via JTAG

Access to Nexus register resources is enabled by loading a single instruction (NEXUS\_ACCESS) into the JTAG Instruction Register (IR). This IR is part of the IEEE 1149.1 TAP controller within the NXDM and NXSI modules.

After the JTAG NEXUS\_ACCESS instruction has been loaded, the JTAG port allows tool/target communications with all Nexus registers according to the map in [Table 51-3](#).

Reading/writing of a Nexus register then requires two (2) passes through the data-scan (DR) path of the JTAG state machine.

- 1. The first pass through the DR selects the Nexus register to be accessed by providing an index (refer to [Table 51-3](#)), and the direction (read/write). This is achieved by loading an 8-bit value into the JTAG data register (DR). [Figure 51-13](#) shows the format of this register.
- 2. The second pass through the DR then shifts the data in or out of the JTAG port, lsb first.
  - a. During a read access, data is latched from the selected Nexus register when the JTAG state machine passes through the capture-DR state.
  - b. During a write access, data is latched into the selected Nexus register when the JTAG state machine passes through the update-DR state.

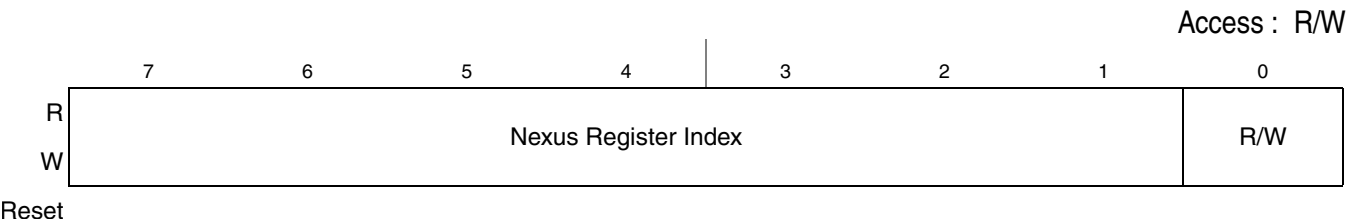


Figure 51-13. JTAG DR for NEXUS Register Access

Table 51-12. DR Read/Write Encoding

| Nexus Register Index | Description |
|----------------------|-------------|
| Read/Write (R/W)     | 0 Read      |

**Table 51-12. DR Read/Write Encoding**

| Nexus Register Index | Description |
|----------------------|-------------|
|                      | 1 Write     |

## 51.4.4 Functional Description

### 51.4.4.1 Enabling NXDM and NXSI Operation

The NXDM (and NXSI) module is enabled by loading a single instruction (ACCESS\_AUX\_TAP\_DMA or ACCESS\_AUX\_TAP\_NXSI) into the JTAG instruction register (IR), and then loading the corresponding OnCE OCMD register with the NEXUS\_ACCESS instruction. After it is enabled, the module is ready to accept control input via the JTAG pins.

The Nexus module is disabled when the JTAG state machine reaches the test-logic-reset state. This state can be reached by the assertion of the JCOMP pin or by cycling through the state machine using the TMS pin. The Nexus module is also disabled if a power-on reset (POR) event occurs.

If the NXDM (and NXSI) module is disabled, no trace output is provided, and the module disables (drive inactive) auxiliary port output pins (MDO[15:0], MSEO[1:0], MCKO). Nexus registers are not be available for reads or writes.

### 51.4.4.2 TCODEs Supported by NXDM and NXSI

The NXDM and NXSI pins allow for flexible transfer operations via public messages. A TCODE defines the transfer format, the number and/or size of the packets to be transferred, and the purpose of each packet. The IEEE-ISTO 5001-2003 standard defines a set of public messages. The NXDM and NXSI blocks currently support the public TCODEs seen in [Table 51-13](#).

**Table 51-13. Public TCODEs Supported**

| Message Name                            | Packet Size<br>Bits |     | Packet Name | Packet Type | Packet Description                                         |
|-----------------------------------------|---------------------|-----|-------------|-------------|------------------------------------------------------------|
|                                         | Min                 | Max |             |             |                                                            |
| Data Trace - Data Write Message         | 6                   | 6   | TCODE       | Fixed       | TCODE number = 5                                           |
|                                         | 4                   | 4   | SRC         | Fixed       | Source processor identifier (multiple Nexus configuration) |
|                                         | 3                   | 3   | DSZ         | Fixed       | Data size (refer to <a href="#">Table 51-15</a> )          |
|                                         | 1                   | 32  | U-ADDR      | Variable    | Unique portion of the data write value                     |
|                                         | 1                   | 64  | DATA        | Variable    | Data write value                                           |
| Data Trace - Data Read Message          | 6                   | 6   | TCODE       | Fixed       | TCODE number = 6                                           |
|                                         | 4                   | 4   | SRC         | Fixed       | Source processor identifier (multiple Nexus configuration) |
|                                         | 3                   | 3   | DSZ         | Fixed       | Data size (refer to <a href="#">Table 51-15</a> )          |
|                                         | 1                   | 32  | U-ADDR      | Variable    | Unique portion of the data read value                      |
|                                         | 1                   | 64  | DATA        | Variable    | Data read value                                            |
| Error Message                           | 6                   | 6   | TCODE       | Fixed       | TCODE number = 8                                           |
|                                         | 4                   | 4   | SRC         | Fixed       | Source processor identifier (multiple Nexus configuration) |
|                                         | 5                   | 5   | ECODE       | Fixed       | Error code (refer to <a href="#">Table 51-14</a> )         |
| Data Trace - Data Write Message w/ Sync | 6                   | 6   | TCODE       | Fixed       | TCODE number = 13 (0xD)                                    |
|                                         | 4                   | 4   | SRC         | Fixed       | Source processor identifier (multiple Nexus configuration) |
|                                         | 3                   | 3   | DSZ         | Fixed       | Data size (refer to <a href="#">Table 51-15</a> )          |
|                                         | 1                   | 32  | F-ADDR      | Variable    | Full access address (leading zero (0) truncated)           |
|                                         | 1                   | 64  | DATA        | Variable    | Data write value                                           |
| Data Trace - Data Read Message w/ Sync  | 6                   | 6   | TCODE       | Fixed       | TCODE number = 14 (0xE)                                    |
|                                         | 4                   | 4   | SRC         | Fixed       | Source processor identifier (multiple Nexus configuration) |
|                                         | 3                   | 3   | DSZ         | Fixed       | Data size (refer to <a href="#">Table 51-15</a> )          |
|                                         | 1                   | 32  | F-ADDR      | Variable    | Full access address (leading zero (0) truncated)           |
|                                         | 1                   | 64  | DATA        | Variable    | Data read value                                            |
| Watchpoint Message                      | 6                   | 6   | TCODE       | Fixed       | TCODE number = 15 (0xF)                                    |
|                                         | 4                   | 4   | SRC         | Fixed       | Source processor identifier (multiple Nexus configuration) |
|                                         | 4                   | 4   | WPHIT       | Fixed       | Number indicating watchpoint sources                       |



**Table 51-14. Error Code (ECODE) Encoding (TCODE = 8)**

| Error Code (ECODE) | Description                                          |
|--------------------|------------------------------------------------------|
| 00000              | Invalid value                                        |
| 00001              | Invalid value                                        |
| 00010              | Data Trace overrun                                   |
| 00011              | Invalid value                                        |
| 00100              | Invalid value                                        |
| 00101              | Invalid access opcode (Nexus Register unimplemented) |
| 00110              | Watchpoint overrun                                   |
| 00111              | Invalid value                                        |
| 01000              | Data Trace and Watchpoint overrun                    |
| 01001-11111        | Invalid value                                        |

**Table 51-15. Data Trace Size (DSZ) Encodings (TCODE = 5, 6, 13, 14)**

| DTM Size Encoding | Transfer Size            |
|-------------------|--------------------------|
| 000               | Byte                     |
| 001               | Halfword (two bytes)     |
| 010               | Word (four bytes)        |
| 011               | Doubleword (eight bytes) |
| 100-111           | Invalid value            |

#### 51.4.4.2.1 Data Trace

This section deals with the data trace mechanism supported by the NXDM and NXSI modules. Data trace is implemented via data write messaging (DWM) and data read messaging (DRM).

#### 51.4.4.2.2 Data Trace Messaging (DTM)

XDM and NXSI data trace messaging is accomplished by snooping the NXDM and NXSI data bus, and storing the information for qualifying accesses (based on enabled features and matching target addresses). The NXDM (and NXSI) module traces all data access that meet the selected range and attributes.

#### NOTE

Data trace is ONLY performed on DMA or SIPI accesses to the system bus.

51.4.4.2.3 DTM Message Formats

The NXDM (and NXSI) block supports five types of DTM Messages - data write, data read, data write synchronization, data read synchronization and error messages.

51.4.4.2.3.1 Data Write and Data Read Messages

The data write and data read messages contain the data write/read value and the address of the write/read access, relative to the previous data trace message. Data write message and data read message information is messaged out in the following format:

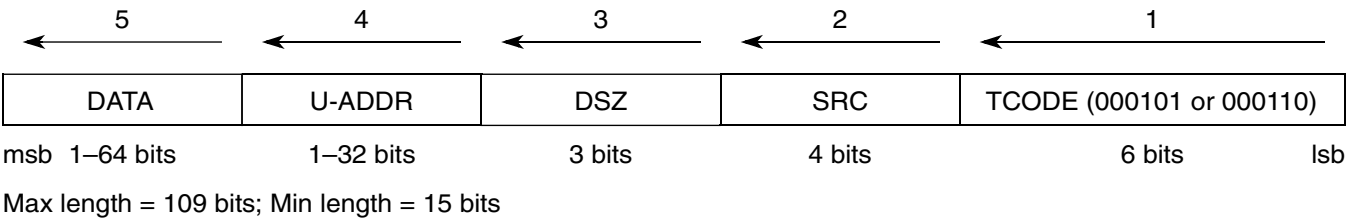


Figure 51-14. Data Write/Read Message Format

51.4.4.2.3.2 DTM Overflow Error Messages

An error message occurs when a new message cannot be queued due to the message queue being full. The FIFO discards incoming messages until it has completely emptied the queue. After it is emptied, an error message is queued. The error encoding indicates which types of messages attempted to be queued while the FIFO was being emptied.

If only a data trace message attempts to enter the queue while it is being emptied, the error message incorporates the data trace only error encoding (00010). If a watchpoint also attempts to be queued while the FIFO is being emptied, then the error message incorporates error encoding (01000).

Error information is messaged out in the following format:

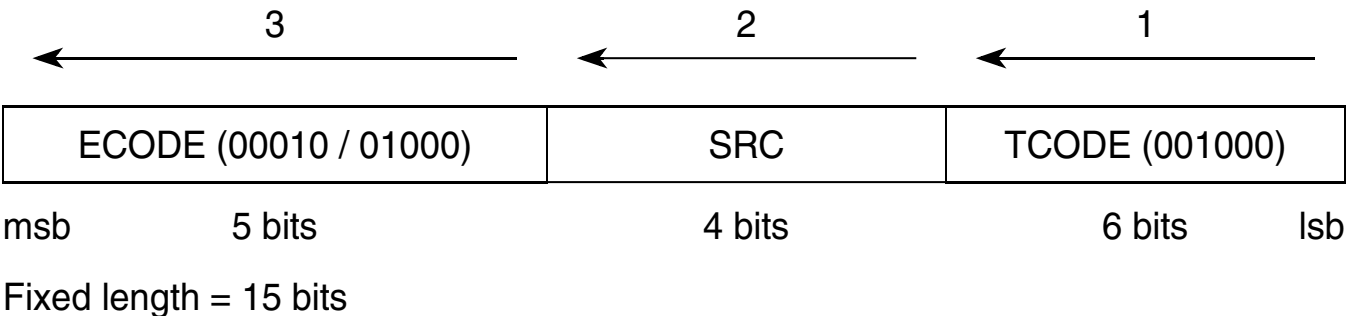


Figure 51-15. Error Message Format

### 51.4.4.2.3.3 Data Trace Synchronization Messages

A data trace write/read w/ sync. message is messaged via the auxiliary port (provided data trace is enabled) for the following conditions (refer to [Table 51-16](#)):

- Initial data trace message upon exit from system reset or whenever data trace is enabled is a synchronization message.
- Upon returning from a low power state, the first data trace message is a synchronization message.
- Upon returning from debug mode, the first data trace message is a synchronization message.
- After occurrence of queue overrun (can be caused by any trace message), the first data trace message is a synchronization message.
- After the periodic data trace counter has expired indicating 255 without-sync data trace messages have occurred since the last with-sync message occurred.
- Upon assertion of the Event In ( $\overline{\text{EVTI}}$ ) pin, the first data trace message is a synchronization message if the eic bits of the dc register have enabled this feature.
- Upon data trace write/read after the previous dtm message was lost due to an attempted access to a secure memory location.
- Upon data trace write/read after the previous dtm message was lost due to a collision entering the fifo between the dtm message and any of the following: error message, or watchpoint message.

Data trace synchronization messages provide the full address (without leading zeros) and insure that development tools fully synchronize with data trace regularly.

Synchronization messages provide a reference address for subsequent DTMs, in which only the unique portion of the data trace address is transmitted. The format for data trace write/read w/ sync. messages is as follows:

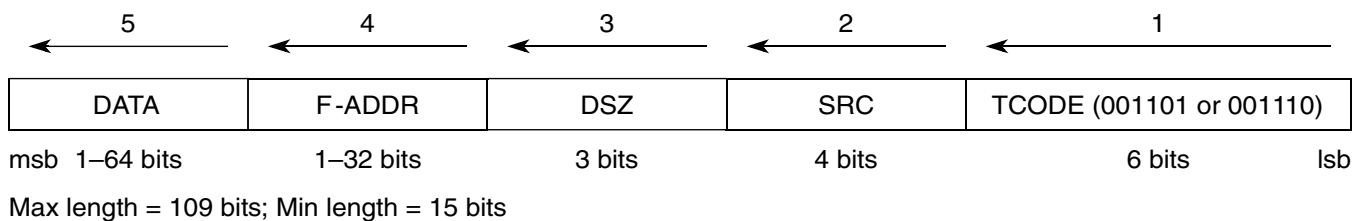


**Figure 51-16. Data Write/Read w/ Sync Message Format**

Exception conditions that result in data trace synchronization are summarized in [Table 51-16](#)."

**Table 51-16. Data Trace Exception Summary**

| Exception Condition                 | Exception Handling                                                                                                                                                                                                                                                                                                                                                                                                |
|-------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| System Reset Negation               | At the negation of JTAG reset (JCOMP), queue pointers, counters, state machines, and registers within the NXDM and NXSI module are reset. If data trace is enabled, the first data trace message is a data write/read w/ sync. message.                                                                                                                                                                           |
| Data Trace Enabled                  | The first data trace message (after data trace has been enabled) is a synchronization message.                                                                                                                                                                                                                                                                                                                    |
| Exit from Low Power/Debug           | Upon exit from a low power mode or debug mode the next data trace message is converted to a data write/read w/ sync. message.                                                                                                                                                                                                                                                                                     |
| Queue Overrun                       | An error message occurs when a new message cannot be queued due to the message queue being full. The FIFO discards messages until it has completely emptied the queue. After it is emptied, an error message is queued. The error encoding indicates the types of messages that attempted to be queued while the FIFO was being emptied. The next DTM message in the queue is a data write/read w/ sync. message. |
| Periodic Data Trace Synchronization | A forced synchronization occurs periodically after 255 data trace messages have been queued. A data write/read w/ sync. message is queued. The periodic data trace message counter then resets.                                                                                                                                                                                                                   |
| Event In                            | If the nexus module is enabled, an EVTI assertion initiates a data trace write/read w/ sync. message upon the next data write/read (if data trace is enabled and the eic bits of the dc register have enabled this feature).                                                                                                                                                                                      |
| Attempted Access to Secure Memory   | Any attempted read or write to secure memory locations temporarily disable data trace & cause the corresponding DTM to be lost. A subsequent read/write queues a data trace read/write with sync. message.                                                                                                                                                                                                        |
| Collision Priority                  | All messages have the following priority: Error → WPM → DTM. A DTM message which attempts to enter the queue at the same time as an error message, or watchpoint message is lost. A subsequent read/write queues a data trace read/write with sync. message.                                                                                                                                                      |

#### 51.4.4.2.4 DTM Operation

##### 51.4.4.2.4.1 Enabling Data Trace Messaging

Data trace messaging can be enabled in one of two ways.

- Setting the DC1[TM] field to enable data trace
- Using the WT[DTS] field to enable data trace on watchpoint hits

##### 51.4.4.2.4.2 DTM Queueing

NXDM and NXSI implements a programmable depth queue for queuing all messages. Messages that enter the queue are transmitted via the auxiliary pins in the order in which they are queued.

**NOTE**

If multiple trace messages must be queued at the same time, watchpoint messages have the highest priority (WPM → DTM).

**51.4.4.2.4.3 Relative Addressing**

The relative address feature is compliant with IEEE-ISTO Nexus 5001-2003 and is designed to reduce the number of bits transmitted for addresses of data trace messages. Relative addressing is the same as described for the NZ7C3.

**51.4.4.2.4.4 Data Trace Windowing**

Data write/read messages are enabled via the RWT1(2) field in the data trace control register (DTC) for each DTM channel. Data trace windowing is achieved via the address range defined by the DTEA and DTSA registers and by the RC1(2) field in the DTC. All eDMA or SIPI initiated read/write accesses that fall inside or outside these address ranges, as programmed, are candidates to be traced.

**51.4.4.2.4.5 System Bus Cycle Special Cases****Table 51-17. System Bus Cycle Special Cases**

| Special Case                               | Action                         |
|--------------------------------------------|--------------------------------|
| System bus cycle aborted (DABORT asserted) | Cycle ignored                  |
| System bus cycle with data error           | Data Trace Message discarded   |
| System bus cycle completed without error   | Cycle captured and transmitted |
| System bus cycle is an instruction fetch   | Cycle ignored                  |

**51.4.4.2.5 Data Trace Timing Diagrams (Eight MDO configuration)**

Data trace timing for the NXDM and NXSI is the same as for the NZ7C3.

**51.4.4.3 Watchpoint Support**

The NXDM and NXSI module provides watchpoint messaging via the auxiliary pins, as defined by IEEE-ISTO 5001-2003.

Watchpoint messages can be generated using the NXDM and NXSI defined internal watchpoints.

51.4.4.3.1 Watchpoint Messaging

Enabling watchpoint messaging is accomplished by setting the watchpoint messaging enable bit, DC1[WEN]. Using the BWC1 and BWC2 registers, two independently controlled internal watchpoints can be initialized. When a DMA or SIPI access address matches on BWA1 or BWA2, a watchpoint message is transmitted.

The Nexus module provides watchpoint messaging using the TCODE. When either of the two possible watchpoint sources asserts, a message is sent to the queue to be messaged out. This message indicates the watchpoint number.

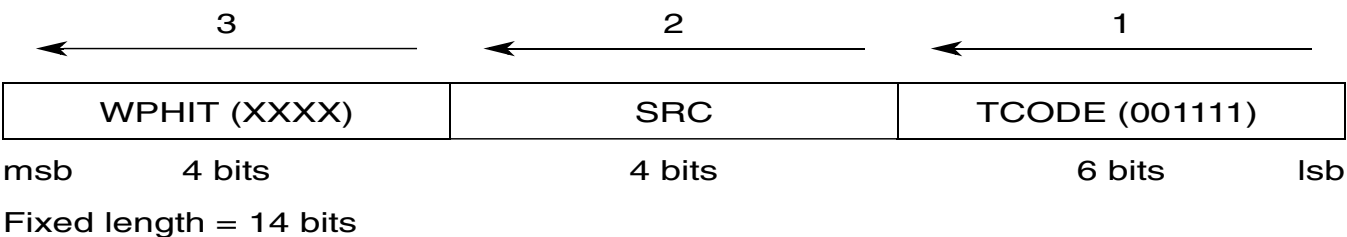


Figure 51-17. Watchpoint Message Format

Table 51-18. Watchpoint Source Description

| Watchpoint Source (4 bits) | Watchpoint Description              |
|----------------------------|-------------------------------------|
| XXX1                       | Invalid value                       |
| XX1X                       | Invalid value                       |
| X1XX                       | Internal Watchpoint #1 (BWA1 match) |
| 1XXX                       | Internal Watchpoint #2 (BWA2 match) |

51.4.4.3.2 Watchpoint Error Message

An error message occurs when a new message cannot be queued due to the message queue being full. The FIFO discards messages until it has completely emptied the queue. After it is emptied, an error message is queued. The error encoding indicates which types of messages attempted to be queued while the FIFO was being emptied.

If only a watchpoint message attempts to enter the queue while it is being emptied, the error message incorporates the watchpoint only error encoding (00110). If a data trace message also attempts to enter the queue while it is being emptied, the error message incorporates error encoding (01000).

Error information is messaged out in the following format (refer to [Figure 51-18](#)).

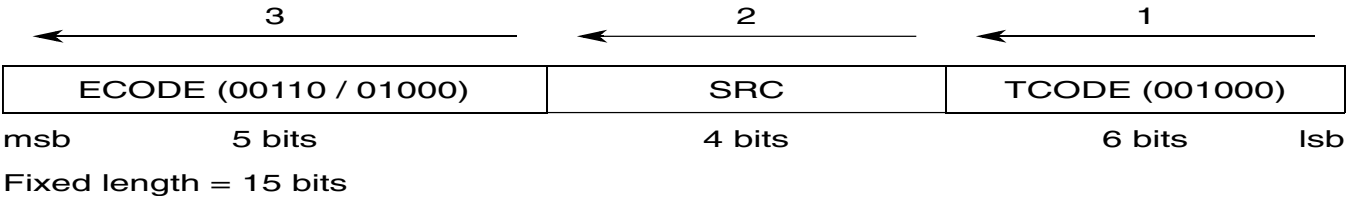


Figure 51-18. Error Message Format





# Chapter 52

## Nexus Port Controller (NPC)

### 52.1 Chip-specific NPC information

The NPC connects to several other Nexus modules that compose the chip's Nexus Development Interface (NDI).

For more information, see the chip-level details about debug and development support.

#### 52.1.1 DID register reset value

The reset value of the [Nexus Device ID \(DID\) Register](#) depends on the chip version.

The PRN, PIN, and DC parameters define the chip-specific values of correspondingly named fields in the DID register. The following table provides their values.

**Table 52-1. Reset values of PRN, DC, and PIN fields of DID register**

| DID field | Device-specific value                                   |
|-----------|---------------------------------------------------------|
| PRN       | 0h for masksets 1N45H and 2N45H<br>1h for maskset 3N45H |
| DC        | 20h                                                     |
| PIN       | 377h                                                    |

#### 52.1.2 NPC MDO pins

The number of available Message Data Out (MDO) pins varies by NPC mode of operation. The following table shows the number of MDO pins available in full-port mode and reduced-port mode.

Table 52-2. MDO pins available by NPC mode of operation

| Mode              | Number of available MDO pins |
|-------------------|------------------------------|
| Full-port mode    | 16                           |
| Reduced-port mode | 12                           |

52.2 Introduction

The following figure is a block diagram of the Nexus Port Controller (NPC) block.

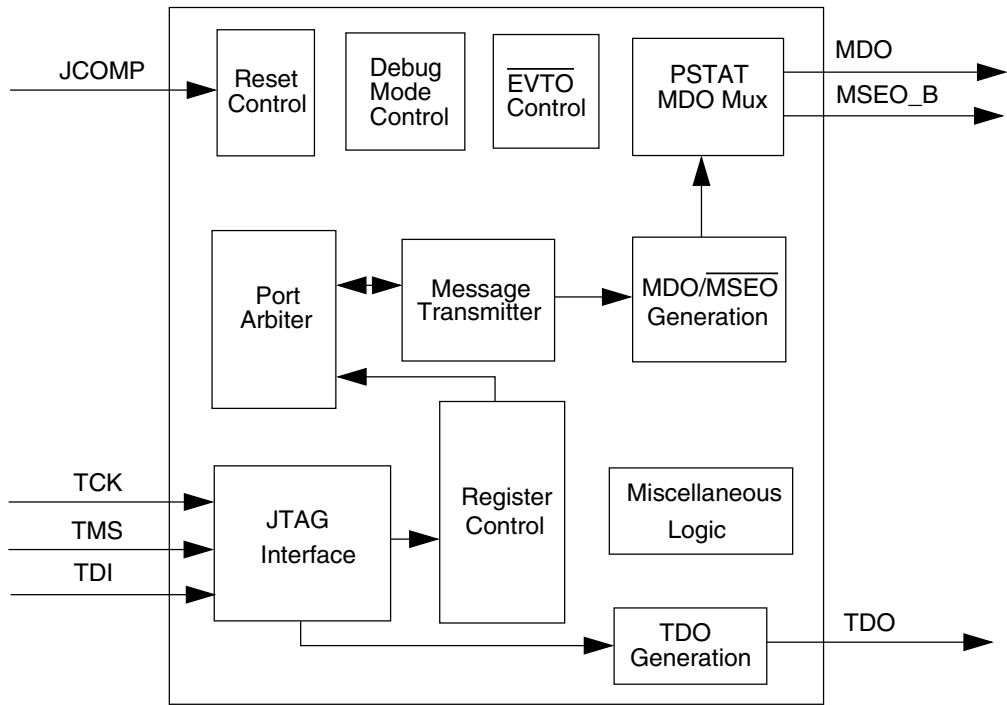


Figure 52-1. Nexus Port Controller Block Diagram

52.2.1 Overview

On a system-on-a-chip device, there are often multiple blocks that require development support. Each of these blocks implements a development interface based on the IEEE-ISTO 5001-2001 standard. The blocks share input and output ports that interface with the development tool. The NPC controls the usage of the input and output port in a manner that allows all the individual development interface blocks to share the port, and appear to the development tool to be a single block.

## 52.2.2 Features

The NPC block performs the following functions:

- Controls arbitration for ownership of the Nexus Output Port
- Nexus Device Identification Register and Messaging
- Generates MCKO enable and frequency division control signals
- Generates a Nexus DDR mode status indication output signal
- Controls sharing of  $\overline{\text{EVTO}}$
- Generates an MCKO clock gating control signal to enable gating of MCKO when the auxiliary output port is idle
- Control of the device-wide debug mode
- Generates asynchronous reset signal for Nexus blocks based on JCOMP input, censorship status, and power-on reset status
- Selectable mode for transmission of processor status information on MDO Auxiliary Output Port pins
- System clock status indication via MDO[0] following power-on reset
- Provides Nexus support for censorship mode

## 52.2.3 Modes of operation

The NPC block uses the JCOMP input, the censorship status signal and an internal power-on reset indication as its primary reset signals. Upon exit of reset, the mode of operation is determined by the Port Configuration Register (PCR) settings.

### 52.2.3.1 Reset

The NPC block is asynchronously placed in reset when either power-on reset is asserted, JCOMP is not set for Nexus access, the device enters censored mode, or the TAP controller state machine is in the Test-Logic-Reset state. Holding TMS high for 5 consecutive rising edges of TCK guarantees entry into the Test-Logic-Reset state regardless of the current TAP controller state. Following negation of power-on reset, the NPC remains in reset until the system clock achieves lock. The NPC is unaffected by other sources of reset. While in reset, the following actions occur:

- The TAP controller is forced into the Test-Logic-Reset state
- The output port pins are negated
- The TDI, TMS, and TCK TAP inputs are ignored (when in power-on reset, censored mode or JCOMP not set for NPC operation only)
- Registers default back to their reset values

### **52.2.3.2 Disabled-Port Mode**

In disabled-port mode, output pin port enable signals are negated, thereby disabling message transmission. Any debug feature that generates messages can not be used. The primary features available are class 1 features and read/write access to the registers. Class 1 features include the ability to trigger a breakpoint event indication through EVTO.

### **52.2.3.3 Full-Port Mode**

Full-port mode (FPM) is entered by asserting the MCKO\_EN and FPM bits in the PCR. In All trace features are enabled or can be enabled by writing the configuration registers via the TAP. The number of MDO pins available is device-specific.

### **52.2.3.4 Reduced-Port Mode**

Reduced-port mode (RPM) is entered by asserting the MCKO\_EN bit and negating the FPM bit in the PCR. All trace features are enabled or can be enabled by writing the configuration registers via the TAP. The number of MDO pins available is device-specific.

### **52.2.3.5 Censored Mode**

In censored mode, it should not be possible to read the contents of internal Flash externally. The NPC and all other Nexus blocks are held in reset when in censored mode , preventing Nexus read/write to memory mapped resources and the transmission of Nexus trace messages.

### 52.2.3.6 Nexus Double Data Rate Mode

Nexus double data rate (DDR) mode is enabled by asserting the DDR\_EN bit in the PCR. DDR mode is only supported for MCKO\_DIV settings of sys\_clk/2, sys\_clk/4 and sys\_clk/8. In double data rate mode, message data is updated between each edge (both rising and falling) of MCKO, effectively doubling the message throughput of single data rate mode.

## 52.3 External signal description

### 52.3.1 Overview

The NPC pin interface provides for the transmission of messages from Nexus blocks to the external development tools and for access to Nexus client registers. The NPC pin definition is outlined in the following table. Pull information for the JTAG signals can be found in the JTAG chapter.

**Table 52-3. NPC Signal Properties**

| Name   | Port      | Function                                | Reset State         |
|--------|-----------|-----------------------------------------|---------------------|
| EVTO_B | Nexus     | Event Out pin                           | 0b1                 |
| JCOMP  | JTAG      | JTAG Compliancy and TAP Sharing Control | —                   |
| MDO    | Auxiliary | Message Data Out pins                   | 0 <sup>1</sup>      |
| MSEO   | Auxiliary | Message Start/End Out pins              | 0b11                |
| TCK    | JTAG      | Test Clock Input                        | —                   |
| TDI    | JTAG      | Test Data Input                         | —                   |
| TDO    | JTAG      | Test Data Output                        | High Z <sup>2</sup> |
| TMS    | JTAG      | Test Mode Select Input                  | —                   |

1. Following a power-on reset, MDO[0] remains asserted until power-on reset is exited and the system clock achieves lock.
2. TDO output buffer enable is negated when the NPC is not in the Shift-IR or Shift-DR states. A weak pull may be implemented on TDO at the SoC level.

### 52.3.2 Detailed signal descriptions

This section describes each of the signals listed in [Table 52-3](#) in more detail. Note that the JTAG test clock (TCK) input from the pin is not a direct input to the NPC. The NPC requires two separate input clocks for TCK clocked logic, one for posedge (rising edge TCK) logic and one for negedge (falling edge TCK) logic. Both clocks are derived from the pin TCK, and generated external to the NPC.

### 52.3.2.1 EVTO\_B — Event Out

Event Out ( $\overline{\text{EVTO}}$ ) is an output pin that is asserted upon breakpoint occurrence to provide breakpoint status indication. The  $\overline{\text{EVTO}}$  output of the NPC is generated based on the values of the individual  $\overline{\text{EVTO}}$  signals from all Nexus blocks that implement the signal.

### 52.3.2.2 JCOMP — JTAG Compliancy

The JCOMP signal provides the ability to share the TAP. The NPC TAP controller remains in reset until it is enabled using the JCOMP signal.

### 52.3.2.3 MDO — Message Data Out

Message Data Out (MDO) are output pins used for uploading OTM, BTM, DTM, and other messages to the development tool. The development tool should sample MDO on the rising edge of MCKO. The width of the MDO bus used is determined by reset configuration.

### 52.3.2.4 MSEO\_B — Message Start/End Out

Message Start/End Out ( $\overline{\text{MSEO}}$ ) is an output pin that indicates when a message on the MDO pins has started, when a variable length packet has ended, or when the message has ended. The development tool should sample  $\overline{\text{MSEO}}$  on the rising edge of MCKO.

### 52.3.2.5 TCK — Test Clock Input

Test Clock Input (TCK) pin is used to synchronize the test logic and control register access through the JTAG port.

### 52.3.2.6 TDI — Test Data Input

Test Data Input (TDI) pin receives serial test instruction and data. TDI is sampled on the rising edge of TCK.

### 52.3.2.7 TDO — Nexus Test Data Output

Test Data Output (TDO) pin transmits serial output for instructions and data. TDO is three-stateable and is actively driven in the SHIFT-IR and SHIFT-DR controller states. TDO is updated on the falling edge of TCK and sampled by the development tool on the rising edge of TCK.

### 52.3.2.8 TMS — Test Mode Select

Test Mode Select Input (TMS) pin is used to sequence the IEEE 1149.1-2001 TAP controller state machine. TMS is sampled on the rising edge of TCK.

## 52.4 Register definition

This section provides a detailed description of the NPC registers accessible to the end user. Individual bit-level descriptions and reset states of the registers are included.

The following table shows the NPC registers by index values. The registers are not memory-mapped and can only be accessed via the TAP. The NPC block does not implement the client select control register because the value does not matter when accessing the registers.

Note that the bypass and instruction registers have no index values. These registers are not accessed in the same manner as Nexus client registers. Refer to the individual register descriptions for more detail.

**Table 52-4. NPC registers**

| Index | Register                          |
|-------|-----------------------------------|
| 0     | Device ID Register (DID)          |
| 127   | Port Configuration Register (PCR) |

### 52.4.1 Register descriptions

This section consists of NPC register descriptions.

### 52.4.1.1 Bypass register

The bypass register is a single-bit shift register path selected for serial data transfer between TDI and TDO when the BYPASS instruction or any unimplemented instructions are active. After entry into the Capture-DR state, the single-bit shift register is set to a logic 0. Therefore, the first bit shifted out after selecting the bypass register is always a logic 0.

### 52.4.1.2 Instruction register

The NPC block uses a 4-bit instruction register as shown in the following table. The instruction register is accessed via the SELECT\_IR\_SCAN path of the tap controller state machine, and allows instructions to be loaded into the block to enable the NPC for register access (NEXUS\_ENABLE) or select the bypass register as the shift path from TDI to TDO (BYPASS or unimplemented instructions).

Instructions are shifted in through TDI while the TAP controller is in the Shift-IR state, and latched on the falling edge of TCK in the Update-IR state. The latched instruction value can only be changed in the Update-IR and Test-Logic-Reset TAP controller states. Synchronous entry into the Test-Logic-Reset state results in synchronous loading of the BYPASS instruction. Asynchronous entry into the Test-Logic-Reset state results in asynchronous loading of the BYPASS instruction. During the Capture-IR TAP controller state, the instruction register is loaded with the value of the previously executed instruction, making this value the register's read value when the TAP controller is sequenced into the Shift-IR state.

|        |                                 |   |   |   |
|--------|---------------------------------|---|---|---|
|        | 3                               | 2 | 1 | 0 |
| R      | Previous Instruction Opcode     |   |   |   |
| W      | Instruction Opcode              |   |   |   |
| Reset: | BYPASS Instruction Opcode (0xF) |   |   |   |



### 52.4.1.3 Nexus Device ID (DID) Register

The device identification register, shown in the following figure, allows the part revision number, design center, part identification number, and manufacturer identity code of the part to be determined through the output port.

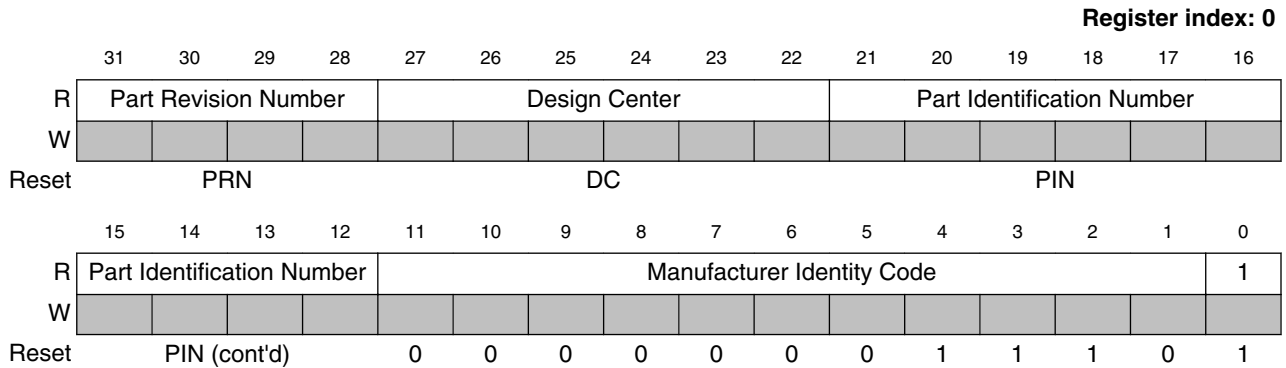


**Table 52-7. DID field descriptions**

| Bit   | Name    | Description                                                                                                                     |
|-------|---------|---------------------------------------------------------------------------------------------------------------------------------|
| 31:28 | PRN     | Part Revision Number<br>These bits contain the revision number of the part                                                      |
| 27:22 | DC      | Design Center<br>These bits indicate the device design center                                                                   |
| 21:12 | PIN     | Part Identification Number<br>These bits contain the part number of the device                                                  |
| 11:1  | MIC     | Manufacturer Identity Code<br>These bits contain the reduced Joint Electron Device Engineering Council (JEDEC) ID for NXP, 0xE. |
| 0     | Bit [0] | IDCODE Register ID<br>Bit [0] identifies this register as the device identification register and not the bypass register        |

## 52.4.1.4 Port Configuration Register (PCR)

| Register index: 127 |          |                 |                 |          |    |    |          |          |          |    |    |    |          |    |    |              |
|---------------------|----------|-----------------|-----------------|----------|----|----|----------|----------|----------|----|----|----|----------|----|----|--------------|
|                     | 31       | 30              | 29              | 28       | 27 | 26 | 25       | 24       | 23       | 22 | 21 | 20 | 19       | 18 | 17 | 16           |
| R                   | FPM      | MCK<br>O_G<br>T | MCK<br>O_E<br>N | MCKO_DIV |    |    | Reserved | DDR_EN   | Reserved | 0  | 0  | 0  | Reserved |    | 0  | 0            |
| W                   |          |                 |                 |          |    |    |          |          |          |    |    |    |          |    |    |              |
| Reset               | 0        | 0               | 0               | 0        | 0  | 0  | 0        | 0        | 0        | 0  | 0  | 0  | 0        | 0  | 0  | 0            |
|                     | 15       | 14              | 13              | 12       | 11 | 10 | 9        | 8        | 7        | 6  | 5  | 4  | 3        | 2  | 1  | 0            |
| R                   | Reserved | 0               | 0               | 0        | 0  | 0  | Reserved | Reserved | 0        | 0  | 0  | 0  | 0        | 0  | 0  | PSTA<br>T_EN |
| W                   |          |                 |                 |          |    |    |          |          |          |    |    |    |          |    |    |              |
| Reset               | 0        | 0               | 0               | 0        | 0  | 0  | 0        | 0        | 0        | 0  | 0  | 0  | 0        | 0  | 0  | 0            |

The PCR, shown in the figure above, is used to select the NPC mode of operation, enable MCKO and select the MCKO frequency, and enable or disable MCKO gating. This register should be configured as soon as the NPC is enabled.

### NOTE

The mode or clock division must not be modified after MCKO has been enabled. Changing the mode or clock division while MCKO is enabled can produce unpredictable results.

**Table 52-9. PCR field descriptions**

| Bit | Name    | Description                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-----|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31  | FPM     | Full Port Mode<br>The value of the FPM bit determines if the auxiliary output port uses the full MDO port or a reduced MDO port to transmit messages.<br>0 A subset of MDO pins are used to transmit messages<br>1 All MDO pins are used to transmit messages                                                                                                                                                                  |
| 30  | MCKO_GT | MCKO Clock Gating Control<br>This bit is used to enable or disable MCKO clock gating. If clock gating is enabled, the MCKO clock is gated when the NPC is in enabled mode but not actively transmitting messages on the auxiliary output port. When clock gating is disabled, MCKO is allowed to run even if no auxiliary output port messages are being transmitted.<br>0 MCKO gating is disabled<br>1 MCKO gating is enabled |
| 29  | MCKO_EN | MCKO Enable<br>This bit enables the MCKO clock to run. When enabled, the frequency of MCKO is determined by the MCKO_DIV field.<br>0 MCKO clock is driven to zero<br>1 MCKO clock is enabled                                                                                                                                                                                                                                   |

*Table continues on the next page...*

**Table 52-9. PCR field descriptions (continued)**

| Bit   | Name     | Description                                                                                                                                                                                                                                                                                            |
|-------|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 28:26 | MCKO_DIV | MCKO Division Factor<br>The value of this signal determines the frequency of MCKO relative to the system clock frequency when MCKO_EN is asserted.<br><a href="#">Table 52-10</a> shows the meaning of MCKO_DIV Values. In this table, SYS_CLK represents the system clock frequency.                  |
| 25    | Reserved |                                                                                                                                                                                                                                                                                                        |
| 24    | DDR_EN   | Double Data Rate Mode Enable<br>This bit enables Nexus double data rate (DDR) mode. In DDR mode, message data is updated between each edge of MCKO, effectively doubling message throughput.<br><b>NOTE:</b> DDR mode should not be used with Aurora mode<br>0 DDR mode disabled<br>1 DDR mode enabled |
| 23    | Reserved |                                                                                                                                                                                                                                                                                                        |
| 19:18 | Reserved |                                                                                                                                                                                                                                                                                                        |
| 15    | Reserved |                                                                                                                                                                                                                                                                                                        |
| 9     | Reserved |                                                                                                                                                                                                                                                                                                        |
| 8     | Reserved |                                                                                                                                                                                                                                                                                                        |
| 0     | PSTAT_EN | Processor Status Mode Enable <sup>1</sup><br>This bit enables processor status (PSTAT) mode. In PSTAT mode, all auxiliary output port MDO pins are used to transmit processor status information, and Nexus messaging is unavailable.<br>0 PSTAT disabled<br>1 PSTAT mode enabled                      |

1. PSTAT Mode is intended for factory processor debug only. The PSTAT\_EN bit should be written to disable PSTAT mode if Nexus messaging is desired. No Nexus messages are transmitted under any circumstances when PSTAT mode is enabled.

**Table 52-10. MCKO\_DIV Values**

| MCKO_DIV[2:0] | MCKO Frequency |
|---------------|----------------|
| 0             | Reserved       |
| 1             | SYS_CLK/2      |
| 2             | Reserved       |
| 3             | SYS_CLK/4      |
| 4             | Reserved       |
| 5             | Reserved       |
| 6             | Reserved       |
| 7             | SYS_CLK/8      |

## 52.5 Functional Description

### 52.5.1 NPC reset configuration

The NPC is placed in disabled mode upon exit of reset. If message transmission via the output port is desired, a write to the PCR is then required to enable the NPC and select the mode of operation. Asserting MCKO\_EN places the NPC in enabled mode and enables MCKO. The frequency of MCKO is selected by writing the MCKO\_DIV field. Asserting or negating the FPM bit selects full-port or reduced-port mode, respectively.

The following table describes the NPC reset configuration options.

**Table 52-11. NPC reset configuration options**

| Power On Reset status asserted? | JCOMP asserted high | MCKO_EN bit of the Port Configuration Register | FPM bit of the Port Configuration Register | Configuration     |
|---------------------------------|---------------------|------------------------------------------------|--------------------------------------------|-------------------|
| yes                             | x                   | X                                              | X                                          | Reset             |
| no                              | no                  | X                                              | X                                          | Reset             |
| no                              | yes                 | 0                                              | X                                          | Disabled          |
| no                              | yes                 | 1                                              | 1                                          | Full-Port Mode    |
| no                              | yes                 | 1                                              | 0                                          | Reduced-Port Mode |

### 52.5.2 Auxiliary Output Port

The auxiliary output port is shared by each of the Nexus modules on the device. The NPC communicates with each of the Nexus modules and arbitrates for access to the port.

#### 52.5.2.1 Output Message Protocol

The protocol for transmitting messages via the auxiliary port is accomplished with the  $\overline{\text{MSEO}}$  functions. The  $\overline{\text{MSEO}}$  pins are used to signal the end of variable-length packets and the end of messages. They are not required to indicate the end of fixed-length packets. MDO and  $\overline{\text{MSEO}}$  are sampled on the rising edge of MCKO.

The following figure illustrates the state diagram for  $\overline{\text{MSEO}}$  transfers. All transitions not included in the figure are reserved, and must not be used.

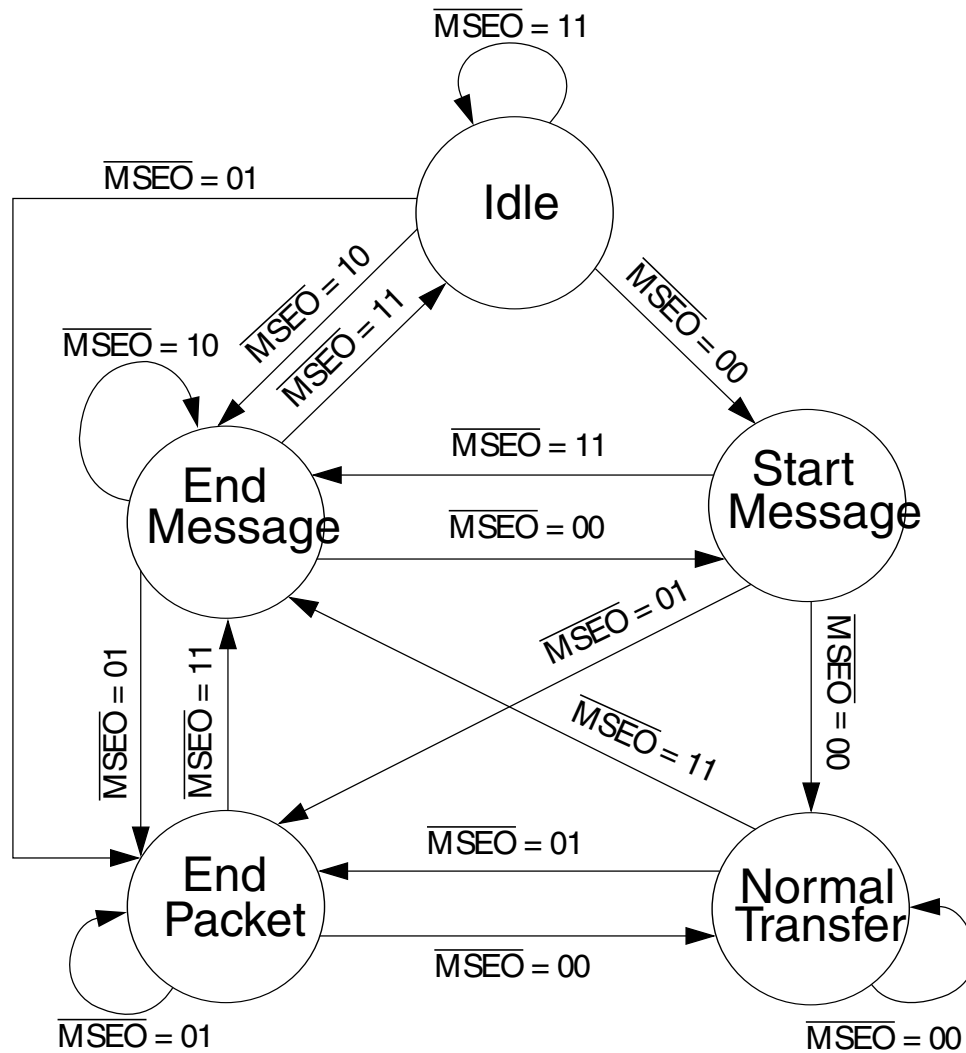


Figure 52-2.  $\overline{\text{MSEO}}$  Transfers (for 2-bit  $\overline{\text{MSEO}}$ )

### 52.5.2.2 Output Messages

In addition to sending out messages generated in other Nexus blocks, the NPC can also output the device ID message contained in the device ID register on the MDO pins. The device ID message can also be sent out serially through TDO.

Figure 52-3 describes the device ID output message that the NPC can transmit on the auxiliary port. The TCODE is the first packet transmitted.

| Message Name      | Min. Packet Size (bits) | Max Packet Size (bits) | Packet Type | Packet Name | Packet Description    |
|-------------------|-------------------------|------------------------|-------------|-------------|-----------------------|
| Device ID Message | 6                       | 6                      | fixed       | TCODE       | Value = 1             |
|                   | 32                      | 32                     | fixed       | ID          | DID register contents |

Figure 52-3. NPC output messages

### 52.5.2.3 Rules of Message

- A variable-sized field within a message must end on a port boundary. (Port boundaries depend on the number of MDO pins active with the current reset configuration.)
- A variable-sized field may start within a port boundary only when following a fixed-length field.
- Super-fields must end on a port boundary.
- When a variable-length field is sized such that it does not end on a port boundary, it is necessary to extend and zero fill the remaining bits after the highest order bit so that it can end on a port boundary.
- Multiple fixed-length packets may start and/or end on a single clock.
- When any packet follows a variable-length packet, it must start on a port boundary.
- The field containing the TCODE number is always transferred out first, followed by subsequent fields of information.
- Within a field, the lowest significant bits are shifted out first. The following figure shows the transmission sequence of a message that is made up of a TCODE followed by two fields.

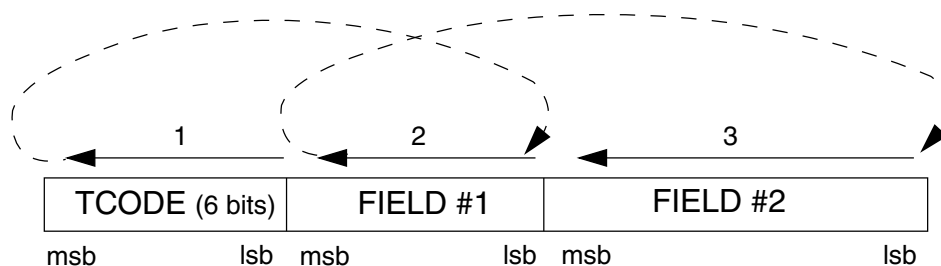


Figure 52-4. Transmission Sequence of Messages

### 52.5.3 IEEE 1149.1-2001 (JTAG) TAP

The NPC block uses the IEEE 1149.1-2001 test access port (TAP) for accessing registers. Each of the individual Nexus blocks on the device implements a TAP controller for accessing its registers as well. TAP signals include TCK, TDI, TMS, and TDO. There may also be other blocks on the MCU that use the TAP and implement a TAP controller.

Refer to the IEEE 1149.1-2001 specification for further detail on electrical and pin protocol compliance requirements.

The NPC implements a Nexus controller state machine that transitions based on the state of the IEEE 1149.1-2001 state machine shown in [Figure 52-6](#). The Nexus controller state machine is defined by the IEEE-ISTO 5001-2001 standard. It is shown in [Figure 52-7](#).

The instructions implemented by the NPC TAP controller are listed in the following table. The value of the NEXUS-ENABLE instruction is 0b0000. Each unimplemented instruction acts like the BYPASS instruction. The size of the NPC instruction register is 4 bits.

**Table 52-12. Implemented Instructions**

| Instruction Name | Private/Public | Opcode | Description                                                                       |
|------------------|----------------|--------|-----------------------------------------------------------------------------------|
| NEXUS-ENABLE     | public         | 0x0    | Activate Nexus controller state machine to read and write NPC registers.          |
| BYPASS           | private        | 0xF    | NPC BYPASS instruction. Also the value loaded into the NPC IR upon exit of reset. |

Data is shifted between TDI and TDO starting with the least significant bit, as illustrated in the following figure. This applies for the instruction register and all Nexus tool-mapped registers.

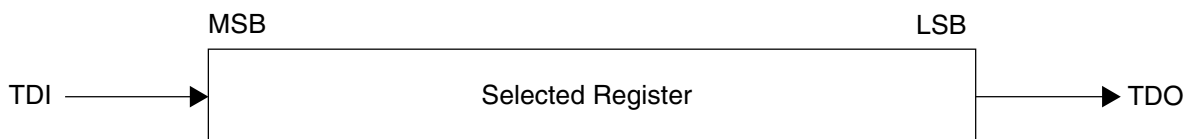
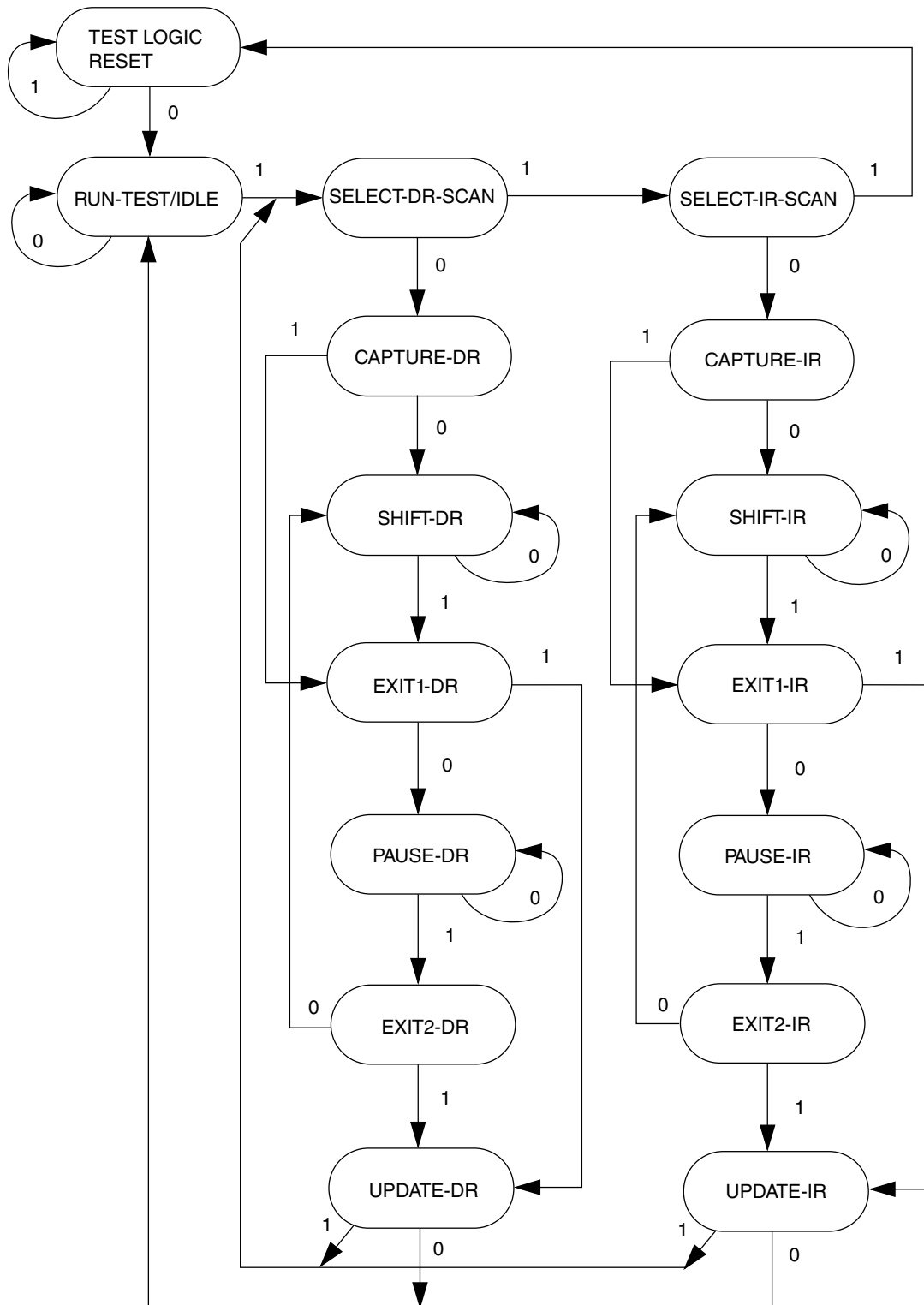


**Figure 52-5. Shifting Data Into Register**

### 52.5.3.1 Enabling the NPC TAP controller

Assertion of the power-on reset signal or entry into censored mode or setting JCOMP to a value other than the NPC enable encoding resets the NPC TAP controller. When not in power-on reset or censored mode, the NPC TAP controller is enabled by driving JCOMP with the NPC enable value and exiting the Test-Logic-Reset state. Loading the NEXUS-ENABLE instruction then grants access to Nexus debug.





NOTE: The value shown adjacent to each state transition in this figure represents the value of TMS at the time of a rising edge of TCK.

**Figure 52-6. IEEE 1149.1-2001 TAP Controller State Machine**

### 52.5.3.2 Retrieving device IDCODE

The Nexus TAP controller does not implement the IDCODE instruction. However, the device identification message can be output by the NPC through the output port or shifted out serially by accessing the Nexus Device ID register through the TAP.

Transmission of the device identification message on the auxiliary output port MDO pins occurs immediately after a write to the PCR, if the NPC is enabled for auxiliary port mode. Transmission of the device identification message serially via the TAP is achieved by performing a read of the register contents as described in [Selecting a Nexus Client Register](#).

### 52.5.3.3 Loading NEXUS-ENABLE Instruction

Access to the NPC registers is enabled when the TAP controller instruction register is loaded with the NEXUS-ENABLE instruction. This instruction is shifted in via the SELECT-IR-SCAN path and loaded in the UPDATE-IR state. At this point, the Nexus controller state machine, shown in [Figure 52-7](#), transitions to the REG\_SELECT state. The Nexus controller has three states: idle, register select, and data access. [Table 52-13](#) illustrates the IEEE 1149.1 sequence to load the NEXUS-ENABLE instruction.

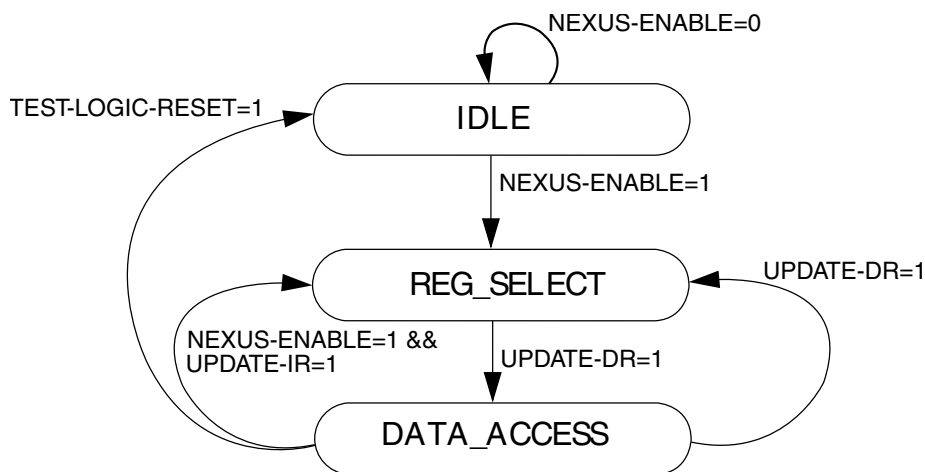


**Figure 52-7. NEXUS Controller State Machine**

**Table 52-13. Loading NEXUS-ENABLE instruction**

| Clock | TMS | IEEE 1149.1 State | Nexus State | Description                                      |
|-------|-----|-------------------|-------------|--------------------------------------------------|
| 0     | 0   | RUN-TEST/IDLE     | IDLE        | IEEE 1149.1-2001 TAP controller in idle state    |
| 1     | 1   | SELECT-DR-SCAN    | IDLE        | Transitional state                               |
| 2     | 1   | SELECT-IR-SCAN    | IDLE        | Transitional state                               |
| 3     | 0   | CAPTURE-IR        | IDLE        | Internal shifter loaded with current instruction |

*Table continues on the next page...*

**Table 52-13. Loading NEXUS-ENABLE instruction  
(continued)**

| Clock  | TMS | IEEE 1149.1 State | Nexus State | Description                                                                                                                   |
|--------|-----|-------------------|-------------|-------------------------------------------------------------------------------------------------------------------------------|
| 4      | 0   | SHIFT-IR          | IDLE        | TDO becomes active, and the IEEE 1149.1-2001 shifter is ready. Shift in all but the last bit of the NEXUS_ENABLE instruction. |
| 3 TCKS |     |                   |             |                                                                                                                               |
| 8      | 1   | EXIT1-IR          | IDLE        | Last bit of instruction shifted in                                                                                            |
| 9      | 1   | UPDATE-IR         | IDLE        | NEXUS-ENABLE loaded into instruction register                                                                                 |
| 10     | 0   | RUN-TEST/IDLE     | REG_SELECT  | Ready to be read/write Nexus registers                                                                                        |

### 52.5.3.4 Selecting a Nexus Client Register

When the NEXUS-ENABLE instruction is decoded by the TAP controller, the input port allows development tool access to all Nexus registers. Each register has a 7-bit address index.

All register access is performed via the SELECT-DR-SCAN path. The Nexus Controller defaults to the REG\_SELECT state when enabled. Accessing a register requires two passes through the SELECT-DR-SCAN path: one pass to select the register and the second pass to read/write the register.

The first pass through the SELECT-DR-SCAN path is used to enter an 8-bit Nexus command consisting of a read/write control bit in the LSB followed by a 7-bit register address index, as illustrated in the following table. The read/write control bit is set to 1 for writes and 0 for reads.

| MSB                  | LSB |
|----------------------|-----|
| 7-bit register index | R/W |

**Figure 52-8. IEEE 1149.1 Controller Command Input**

The second pass through the SELECT-DR-SCAN path is used to read or write the register data by shifting in the data (LSB first) during the SHIFT-DR state. When reading a register, the register value is loaded into the JTAG shifter during the CAPTURE-DR state. When writing a register, the value is loaded from the IEEE 1149.1-2001 shifter to the register during the UPDATE-DR state. When reading a register, there is no requirement to shift out the entire register contents. Shifting may be terminated once the required number of bits have been acquired.

The following table illustrates a sequence which writes a 32-bit value to a register.

**Table 52-14. Write to a 32-Bit Nexus Client Register**

| Clock   | TMS | IEEE 1149.1 State | Nexus State | Description                                                                                          |
|---------|-----|-------------------|-------------|------------------------------------------------------------------------------------------------------|
| 0       | 0   | RUN-TEST/IDLE     | REG_SELECT  | IEEE 1149.1-2001 TAP controller in idle state                                                        |
| 1       | 1   | SELECT-DR-SCAN    | REG_SELECT  | First pass through SELECT-DR-SCAN path                                                               |
| 2       | 0   | CAPTURE-DR        | REG_SELECT  | Internal shifter loaded with current value of controller command input.                              |
| 3       | 0   | SHIFT-DR          | REG_SELECT  | TDO becomes active, and write bit and 6 bits of register index shifted in.                           |
| 7 TCKs  |     |                   |             |                                                                                                      |
| 11      | 1   | EXIT1-DR          | REG_SELECT  | Last bit of register index shifted into TDI                                                          |
| 12      | 1   | UPDATE-DR         | REG_SELECT  | Controller decodes and selects register                                                              |
| 13      | 1   | SELECT-DR-SCAN    | DATA_ACCESS | Second pass through SELECT-DR-SCAN path                                                              |
| 14      | 0   | CAPTURE-DR        | DATA_ACCESS | Internal shifter loaded with current value of register                                               |
| 15      | 0   | SHIFT-DR          | DATA_ACCESS | TDO becomes active, and outputs current value of register while new value is shifted in through TDI  |
| 31 TCKs |     |                   |             |                                                                                                      |
| 47      | 1   | EXIT1-DR          | DATA_ACCESS | Last bit of current value shifted out TDO. Last bit of new value shifted in TDI.                     |
| 48      | 1   | UPDATE-DR         | DATA_ACCESS | Value written to register                                                                            |
| 49      | 0   | RUN-TEST/IDLE     | REG_SELECT  | Controller returned to idle state. It could also return to SELECT-DR-SCAN to write another register. |

## 52.5.4 Nexus JTAG Port Sharing

Each of the individual Nexus blocks on the device implements a TAP controller for accessing its registers. When Nexus has ownership of the TAP, only the block whose NEXUS-ENABLE instruction is loaded has control of the TAP. This allows the interface to all of these individual TAP controllers to appear to be a single port from outside the device. If no register is selected as the shift path for a Nexus block, that block acts like a single-bit shift register, or bypass register.

## 52.5.5 MCKO Control

MCKO is an output clock to the development tools used for the timing of  $\overline{\text{MSE0}}$  and MDO pin functions. MCKO is derived from the system clock and its frequency is determined by the value of the MCKO\_DIV field in the PCR. Possible operating frequencies include system clock, one-half system clock, one-quarter system clock, and one-eighth system clock speed.

The NPC also generates an MCKO clock gating control output signal. This output can be used by the MCKO generation logic to gate the transmission of MCKO when the auxiliary port is enabled but not transmitting messages. The setting of the MCKO\_GT bit inside the PCR determines whether or not MCKO gating control is active. The MCKO\_GT bit resets to a logic 0. In this state gating of MCKO is disabled. To enable gating of MCKO, the MCKO\_GT bit in the PCR is written to a logic 1.

### 52.5.6 $\overline{\text{EVTO}}$ Sharing

The NPC block controls sharing of the  $\overline{\text{EVTO}}$  output between all Nexus clients that produce an  $\overline{\text{EVTO}}$  signal. The NPC assumes incoming  $\overline{\text{EVTO}}$  signals will be asserted for one system clock period. After receiving a single clock period of asserted  $\overline{\text{EVTO}}$  from any Nexus client, the NPC latches the result, and drives  $\overline{\text{EVTO}}$  for one MCKO period on the following clock. When there is no active MCKO, such as in disabled mode, the NPC drives  $\overline{\text{EVTO}}$  for two system clock periods.  $\overline{\text{EVTO}}$  sharing is active as long as the NPC is not in reset.

### 52.5.7 Nexus Reset Control

The JCOMP input that is used as the primary reset signal for the NPC is also used by the NPC to generate a single-bit reset signal for other Nexus blocks. If JCOMP is negated, an internal reset is asserted, indicating that all Nexus modules should be held in reset. Internal Nexus reset is also asserted when the device is in censored mode.

### 52.5.8 System Clock Locked Indication

Following a power-on reset, MDO[0] can be monitored to provide the lock status of the system clock. MDO[0] is driven to a logic 1 until the system clock achieves lock after exiting power-on reset. Once the system clock is locked, MDO[0] is negated and tools may begin Nexus configuration. Loss of lock conditions that occur subsequent to the exit of power-on reset and the initial lock of the system clock do not cause a Nexus reset, and therefore do not result in MDO[0] driven high.

## 52.6 Initialization/Application Information

### 52.6.1 Accessing NPC tool-mapped registers

To initialize the TAP for Nexus register accesses, the following sequence is required:

1. Enable the Nexus TAP controller
2. Load the TAP controller with the NEXUS-ENABLE instruction

To write control data to NPC tool-mapped registers, the following sequence is required:

1. Write the 7-bit register index and set the write bit to select the register with a pass through the SELECT-DR-SCAN path in the TAP controller state machine.
2. Write the register value with a second pass through the SELECT-DR-SCAN path. Note that the prior value of this register is shifted out during the write.

To read status and control data from NPC tool-mapped registers, the following sequence is required:

1. Write the 7-bit register index and clear the write bit to select register with a pass through SELECT-DR-SCAN path in the TAP controller state machine.
2. Read the register value with a second pass through the SELECT-DR-SCAN path. Data shifted in is ignored.

See the IEEE-ISTO 5001-2001 standard for more detail.

# Chapter 53

## Nexus eTPU Development Interface (NDEDI)

### 53.1 Chip-specific NDEDI information

This chip contains two Nexus eTPU Development Interface modules.

- A dual NDEDI covers eTPU\_A and eTPU\_B. The NDEDI's ENGINE1 corresponds to eTPU\_A and ENGINE2 corresponds to eTPU\_B.
- A single Nexus eTPU Development Interface (NSEDI) covers eTPU\_C. The single interface has the same features as the NDEDI but is for a single eTPU engine.

Each interface also includes a coherent dual-parameter controller (CDC) for coherent access to eTPU parameters. In each interface, the eTPU modules and CDC unit appear as separate Nexus clients.

#### 53.1.1 NDEDI/NSEDI device-specific parameters

The following table provides the chip-specific parameter values of the NDEDI module and NSEDI module. Each parameter has a placeholder "variable" name that appears in the detailed NDEDI description.

**Table 53-1. Device-specific parameters**

| Parameter                  | Variable name | Description                                                                                                                          | NDEDI value | NSEDI value |
|----------------------------|---------------|--------------------------------------------------------------------------------------------------------------------------------------|-------------|-------------|
| Full-Port Mode MDO pins    | N             | Number of MDO pins used for message transmission in full-port mode: This value is the same for all Nexus blocks sharing the port.    | 16d         | 16d         |
| Reduced-Port Mode MDO pins | J             | Number of MDO pins used for message transmission in reduced-port mode: This value is the same for all Nexus blocks sharing the port. | 12d         | 12d         |
| SRC Packet size            | K             | Number of bits used for the SRC packet in messages                                                                                   | 4d          | 4d          |

*Table continues on the next page...*

**Table 53-1. Device-specific parameters (continued)**

| Parameter                  | Variable name   | Description                                                                 | NDEDI value | NSEDI value |
|----------------------------|-----------------|-----------------------------------------------------------------------------|-------------|-------------|
| HIST Packet size           | H               | Number of bits used for the HIST packet in messages                         | 26d         | 26d         |
| Queue size                 | Q_SIZE          | Depth of the event queue used to store events before transmission           | 32d         | 32d         |
| Stall Threshold            | STALL_THRESHOLD | Number of spots within the queue used as a threshold to delay the processor | 7d          | 7d          |
| ENGINE1 SRC value          | ENGINE1_SRC     | SRC packet value for messages generated by ENGINE1                          | 2h          | Ah          |
| ENGINE2 SRC value          | ENGINE2_SRC     | SRC packet value for messages generated by ENGINE2                          | 3h          | —           |
| CDC SRC value              | CDC_SRC         | SRC packet value for messages generated by the CDC                          | 4h          | Ch          |
| Manufacturer ID            | DID_MIC         | Manufacturer code used in the DID register                                  | Eh          | Eh          |
| Design Center ID           | DID_DC          | Design center code used in the DID register                                 | 20h         | 20h         |
| Part Revision Number       | DID_PRN         | Part revision number used in the DID register                               | 4h          | 4h          |
| Part Identification Number | DID_PIN         | Part identification number used in the DID register                         | 224h        | 124h        |

## 53.2 Introduction

The Nexus ETPU Development Interface (NDEDI) provides real-time development capabilities for the ETPU system including two engines and the coherent dual-parameter controller (CDC) in compliance with the IEEE-ISTO 5001-2002 standard. The NDEDI interfaces to the ETPU via dedicated trace busses. The main development features supported are breakpoint/watchpoint configuration, debug mode register access, enter debug mode at reset negation or during normal execution, single stepping of instructions, branch trace, data trace, ownership trace, and watchpoint trace. Combined, these features make the interface for each engine compliant with Class 3 of the IEEE-ISTO 5001-2002 standard.

Both engines have their own Nexus register sets that allows trace to be set up independently for each of them. The only exception to this is the data trace address range registers that are shared. All the registers are accessed via the JTAG port. Output messages between the engines and other sources are differentiated by the value of the SRC packet in the output messages.



The NDEDI block is combined with similar blocks for other intelligent peripherals, a read/write access block, and a Nexus Port Controller block to provide the complete IEEE-ISTO 5001-2002 interface for a system-on-a-chip design. The JTAG port and Nexus auxiliary port pins are shared by all of the Nexus based blocks on the chip.

### 53.3 Features

The NDEDI block is compliant with the IEEE-ISTO 5001-2002 standard, and implements the following features:

- Full duplex pin interface for medium and high visibility throughput
  - Pin interface shared among all Nexus-based development interfaces on the device
  - One of two modes selected during reset: full-port mode (FPM) and reduced-port mode (RPM)
- Auxiliary Output Port
  - 1 MCKO (Message Clock Out) pin
  - MDO (Message Data Out) port (port size is SoC specific)
  - 2  $\overline{\text{MSEO}}$  (Message Start/End Out) pins
  - 1  $\overline{\text{EVTO}}$  (Event Out) pin
- Auxiliary Input Port
  - 1  $\overline{\text{EVTI}}$  (Event In) pin
- IEEE 1149.1 (JTAG) Test Access Port (TAP)
  - Support for optional Multi-JTAG TAP Linking Module (TLM)
  - 4 pins (TDI, TDO, TMS, and TCK)
  - Reset input  $\overline{\text{TRST}}$  driven by either the Nexus Port Controller or an external pin
- ETPU Development Support
  - IEEE-ISTO 5001-2002 standard class 3 compliant for both engines
  - Read/Write registers in debug mode
  - Ability to enter debug mode at reset negation or during normal execution

- Ability to single step an instruction and re-enter debug mode
- Breakpoint and Watchpoint Configuration
- Execute external microcode instructions in debug mode providing indirect read/write accesses to ETPU registers
- Read the microprogram counter in debug mode
- Change the microprogram counter by executing external microcode instructions in debug mode
- Data Trace via Data Write Messaging (DWM) and Data Read Messaging (DRM). This allows the development tool to trace reads and writes to selected shared parameter RAM (SPRAM) address ranges
  - Four data trace windows with programmable address ranges and access attributes are provided. Data trace windowing reduces the requirement on the auxiliary port bandwidth by constraining the number of trace locations. The four trace window address ranges are shared among the dual engines and the ETPU Coherent Dual-Parameter Controller (CDC)
- Ownership Trace via Ownership Trace Messaging (OTM). OTM provides visibility of which channel is being serviced. An ownership trace message is transmitted to indicate when a new channel service request is scheduled, allowing the development tools to trace task flow. A special OTM is sent when the engine enters in idle, meaning that all requests were serviced and no new requests are yet scheduled
- Program Trace via Branch Trace Messaging (BTM). BTM displays program flow discontinuities (start, jump, return, etc....), allowing the development tool to interpolate what transpires between the discontinuities. Thus static code may be traced
  - Public messages are used for jump, and return instructions
  - Private messages are used to indicate special ETPU cases not covered by public messages. These messages include the writes to the channel register, the start of a channel service and the trace enabling after a watchpoint hit
  - Branch/Predicate History is kept on direct branches (taken and not taken) and predicated instructions (executed or not executed) to minimize the number of messages transmitted
- Watchpoint Messaging (WPM) via the auxiliary port. WPM provides visibility of the occurrence of the ETPUs' watchpoints

- Event Queue Interface. The Event Queue array is located external to the NDEDI block to separate hard and soft IP. The depth of the Event Queue is specified in the SoC Guide
- Read/Write Access functions are provided on the device level by a block separate from the NDEDI block

## 53.4 Modes of operation

### 53.4.1 Reset

The NDEDI block is placed in reset when Nexus is reset, or the TAP controller state machine is in the Test-Logic-Reset state. Holding TMS high for 5 consecutive rising edges of TCK guarantees entry into the Test-Logic-Reset state regardless of the current TAP controller state. Nexus reset results in asynchronous entry into the reset state. The NDEDI is unaffected by other sources of reset. While in reset, the following actions occur:

- The TAP controller is forced into the Test-Logic-Reset state
- The auxiliary output port pins are negated
- The auxiliary output port enable outputs are negated
- The TDI, TMS, and TCK TAP inputs are ignored
- Registers default back to their reset values

Note that the NDEDI is not reset by the system reset signal. The system reset signal is used though to make ETPU entering in debug mode from reset condition and to synchronize Program Trace and Data Trace messages.

### 53.4.2 Disabled-port mode

In disabled-port mode, auxiliary output pin port enable signals are negated, thereby disabling message transmission. Thus, the NDEDI class 3 features are not available in this mode. The primary features available are class 1 features and read/write access to the registers.

### 53.4.3 Full-port mode

Full-port mode (FPM) is determined by the chip Nexus configuration. See the chip-specific NPC information. In full-port mode, the block is enabled and the FPM port enable is asserted indicating that all available MDO pins are to be used for message transmission. All trace features are enabled or can be enabled by writing the configuration registers via the TAP. The number of MDO pins available is device-specific. Refer to the SoC guide for this information.

### 53.4.4 Reduced-port mode

Reduced-port mode (RPM) is determined by the chip Nexus configuration. See the chip-specific NPC information. In reduced-port mode, the block is enabled and the RPM port enable is asserted indicating that only a subset of the MDO pins are to be used for message transmission. All trace features are enabled or can be enabled by writing the configuration registers via the TAP. The number of MDO pins available is device-specific. Refer to the SoC guide for this information.

### 53.4.5 Stop mode

The NDEDI supports the IPI Green-Line Interface stop mode mechanism. When a request is made to enter stop mode, the NDEDI block waits for the ETPU to acknowledge the stop condition so no trace information is lost. The NDEDI then completes transmitting all pending messages, emptying the Event Queue. Finally, NDEDI transmits a debug status message to indicate it is entering stop mode, and if Program Trace is enabled it will send a correlation message. Refer to [Functional description](#) for further information about the messages. After this is completed, the request to enter stop mode is acknowledged. No breakpoint requests are sent to the engines while they are in stop mode. After the acknowledgement, the system clock input may be shut off by the clock driver on the device. While the clocks are shut off, the development tool must not access NDEDI registers via the JTAG test access port.

## 53.5 External signal description

The following section describes the External Signal description.

## 53.5.1 Overview

The NDEDI pin provides interface for the transmission of messages and for accessing Nexus registers. The NDEDI pin definition is outlined in the following table.

**Table 53-2. NDEDI signal properties**

| Name                            | Port      | Function                         | Reset State | Pull <sup>1</sup> |
|---------------------------------|-----------|----------------------------------|-------------|-------------------|
| EVT $\overline{\text{I}}$       | Auxiliary | Event In pin                     | —           | Up                |
| EVT $\overline{\text{O}}$       | Auxiliary | Event Out pin                    | 1           | —                 |
| MCKO                            | Auxiliary | Message Clock Out pin (from NPC) | —           | —                 |
| MDO[J-1:0]<br>or<br>MDO[N-1:0]  | Auxiliary | Message Data Out pins            | 0           | —                 |
| MSE $\overline{\text{O}}$ [1:0] | Auxiliary | Message Start/End Out pins       | 0b11        | —                 |
| TCK                             | TAP       | Test Clock Input                 | —           | —                 |
| TDI                             | TAP       | Test Data Input                  | —           | Up                |
| TDO                             | TAP       | Test Data Output                 | 0           | —                 |
| TMS                             | TAP       | Test Mode Select Input           | —           | Up                |
| TRST                            | TAP       | Test Reset Input                 | —           | Up                |

1. The pull is not implemented in this block. Refer to the SoC guide for more details.

## 53.5.2 Detailed signal descriptions

This section describes each of the signals listed in the preceding table in more detail.

### 53.5.2.1 $\overline{\text{EVTI}}$ - Event In

Event In ( $\overline{\text{EVTI}}$ ) is used to initiate program and data trace synchronization messages or to generate a breakpoint.  $\overline{\text{EVTI}}$  is edge-sensitive for synchronization and breakpoint generation.

### 53.5.2.2 $\overline{\text{EVTO}}$ - Event Out

Event Out ( $\overline{\text{EVTO}}$ ) is an output pin to development tools comprising exact timing for a single watchpoint or breakpoint occurrence.  $\overline{\text{EVTO}}$  is asserted for 1 MCKO clock<sup>1</sup>.

1. NDEDI asserts The EVTO indication for 1 system clock.

### 53.5.2.3 MCKO - Message Clock Out

Message Clock Out (MCKO) is an output clock to the development tools used for the timing of MDO and  $\overline{\text{MSEO}}$  pin functions. The frequency of MCKO is determined by the NPC block. Refer to the NPC BG for more information.

The MCKO signal is generated external to the NDEDI block but is included in this document for functional clarity. An input is provided to this block to keep logic synchronized to MCKO.

### 53.5.2.4 MDO - Message Data Out

Message Data Out (MDO) are output pins used for uploading messages to the development tool. The development tool must sample MDO on the rising edge of MCKO. The width of the MDO bus is determined by the Nexus full port mode configuration. See the chip-specific NPC information for further information.

### 53.5.2.5 MSEO[1:0] - Message Start/End Out

Message Start/End Out ( $\overline{\text{MSEO}}$ ) is an output pin that indicates when a message on the MDO pins has started, when a variable length packet has ended, or when the message has ended. The development tool must latch  $\overline{\text{MSEO}}$  on the rising edge of MCKO.

### 53.5.2.6 TCK - Test Clock Input

Test Clock Input (TCK) is an input pin used to synchronize the test logic and control register access through the JTAG port.

### 53.5.2.7 TDI - Test Data Input

Test Data Input (TDI) is an input pin that receives serial test instructions and data. TDI is sampled on the rising edge of TCK.

### 53.5.2.8 TDO - Test Data Output

Test Data Output (TDO) is an output pin that transmits serial output for test instructions and data. TDO is actively driven in the 'SHIFT-IR' and 'SHIFT-DR' states of the JTAG state machine. TDO changes on the falling edge of TCK and is sampled on the rising edge of TCK.

### 53.5.2.9 TMS - Test Mode Select

Test Mode Select (TMS) is an input pin used to sequence the JTAG state machine. TMS is sampled on the rising edge of TCK.

### 53.5.2.10 $\overline{\text{TRST}}$ - Test Reset Input

The Test Reset Input is used to asynchronously reset the NDEDI block and initialize the JTAG controller. This signal can be generated in the Nexus Port Controller or driven by an external pin. Refer to the SoC Guide for more information.

## 53.6 Register definition

This section provides a detailed description of all NDEDI registers accessible to the development tool. Individual bit-level descriptions and reset states of each register are included.

The following table shows the NDEDI registers by client select and index values. These registers are not memory-mapped and can only be accessed via the TAP.

**Table 53-3. NDEDI Registers**

| CS                               | Index | Register                                                     |
|----------------------------------|-------|--------------------------------------------------------------|
| Client-Independent Registers     |       |                                                              |
| X                                | 0     | Device Identity (DID)                                        |
| X                                | 1     | Client Select Control (CSC)                                  |
| ENGINE1 Control/Status Registers |       |                                                              |
| ENGINE1_SRC                      | 2     | ENGINE1 Development Control (NDEDI_ENGINE1_DC)               |
| ENGINE1_SRC                      | 4     | ENGINE1 Development Status (NDEDI_ENGINE1_DS)                |
| ENGINE1_SRC                      | 11    | ENGINE1 Watchpoint Trigger (NDEDI_ENGINE1_WT)                |
| ENGINE1_SRC                      | 13    | ENGINE1 Data Trace Control (NDEDI_ENGINE1_DTC)               |
| ENGINE1_SRC                      | 22    | ENGINE1 Breakpoint/Watchpoint Control 1 (NDEDI_ENGINE1_BWC1) |
| ENGINE1_SRC                      | 23    | ENGINE1 Breakpoint/Watchpoint Control 2 (NDEDI_ENGINE1_BWC2) |

*Table continues on the next page...*

**Table 53-3. NDEDI Registers (continued)**

| CS                                                  | Index | Register                                                     |
|-----------------------------------------------------|-------|--------------------------------------------------------------|
| ENGINE1_SRC                                         | 24    | ENGINE1 Breakpoint/Watchpoint Control 3 (NDEDI_ENGINE1_BWC3) |
| ENGINE1_SRC                                         | 30    | ENGINE1 Breakpoint/Watchpoint Address 1 (NDEDI_ENGINE1_BWA1) |
| ENGINE1_SRC                                         | 31    | ENGINE1 Breakpoint/Watchpoint Address 2 (NDEDI_ENGINE1_BWA2) |
| ENGINE1_SRC                                         | 38    | ENGINE1 Breakpoint/Watchpoint Data 1 (NDEDI_ENGINE1_BWD1)    |
| ENGINE1_SRC                                         | 39    | ENGINE1 Breakpoint/Watchpoint Data 2 (NDEDI_ENGINE1_BWD2)    |
| ENGINE1_SRC                                         | 64    | ENGINE1 Program Trace Channel Enable (NDEDI_ENGINE1_PTCE)    |
| ENGINE1_SRC                                         | 69    | ENGINE1 Microinstruction Debug Register (NDEDI_ENGINE1_INST) |
| ENGINE1_SRC                                         | 70    | ENGINE1 Microinstruction Program Counter (NDEDI_ENGINE1_MPC) |
| ENGINE1_SRC                                         | 71    | ENGINE1 Channel Flag Status Register(NDEDI_ENGINE2_CSFR)     |
| ENGINE2 Control/Status Registers                    |       |                                                              |
| ENGINE2_SRC                                         | 2     | ENGINE2 Development Control (NDEDI_ENGINE2_DC)               |
| ENGINE2_SRC                                         | 4     | ENGINE2 Development Status (NDEDI_ENGINE2_DS)                |
| ENGINE2_SRC                                         | 11    | ENGINE2 Watchpoint Trigger (NDEDI_ENGINE2_WT)                |
| ENGINE2_SRC                                         | 13    | ENGINE2 Data Trace Control (NDEDI_ENGINE2_DTC)               |
| ENGINE2_SRC                                         | 22    | ENGINE2 Breakpoint/Watchpoint Control 1 (NDEDI_ENGINE2_BWC1) |
| ENGINE2_SRC                                         | 23    | ENGINE2 Breakpoint/Watchpoint Control 2 (NDEDI_ENGINE2_BWC2) |
| ENGINE2_SRC                                         | 24    | ENGINE2 Breakpoint/Watchpoint Control 3 (NDEDI_ENGINE2_BWC3) |
| ENGINE2_SRC                                         | 30    | ENGINE2 Breakpoint/Watchpoint Address 1 (NDEDI_ENGINE2_BWA1) |
| ENGINE2_SRC                                         | 31    | ENGINE2 Breakpoint/Watchpoint Address 2 (NDEDI_ENGINE2_BWA2) |
| ENGINE2_SRC                                         | 38    | ENGINE2 Breakpoint/Watchpoint Data 1 (NDEDI_ENGINE2_BWD1)    |
| ENGINE2_SRC                                         | 39    | ENGINE2 Breakpoint/Watchpoint Data 2 (NDEDI_ENGINE2_BWD2)    |
| ENGINE2_SRC                                         | 64    | ENGINE2 Program Trace Channel Enable (NDEDI_ENGINE2_PTCE)    |
| ENGINE2_SRC                                         | 69    | ENGINE2 Microinstruction Debug Register (NDEDI_ENGINE2_INST) |
| ENGINE2_SRC                                         | 70    | ENGINE2 Microinstruction Program Counter (NDEDI_ENGINE2_MPC) |
| ENGINE2_SRC                                         | 71    | ENGINE2 Channel Flag Status Register(NDEDI_ENGINE2_CSFR)     |
| ETPU CDC Control/Status Registers                   |       |                                                              |
| CDC_SRC                                             | 13    | CDC Data Trace Control (NDEDI_CDC_DTC)                       |
| ENGINE1/ENGINE2/CDC Shared Control/Status Registers |       |                                                              |
| ENGINE1_SRC or<br>ENGINE2_SRC or<br>CDC_SRC         | 65    | NDEDI Data Trace Address Range 0 (NDEDI_DTAR0)               |
| ENGINE1_SRC or<br>ENGINE2_SRC or<br>CDC_SRC         | 66    | NDEDI Data Trace Address Range 1 (NDEDI_DTAR1)               |
| ENGINE1_SRC or<br>ENGINE2_SRC or<br>CDC_SRC         | 67    | NDEDI Data Trace Address Range 2 (NDEDI_DTAR2)               |
| ENGINE1_SRC or<br>ENGINE2_SRC or<br>CDC_SRC         | 68    | NDEDI Data Trace Address Range 3 (NDEDI_DTAR3)               |



### 53.6.1 Register descriptions

This section consists of NDEDI register descriptions. The ENGINE1 and ENGINE2 registers are shown in the preceding table. To simplify the document, the description of registers with the same content is only given once instead of being repeated for each ENGINE. This is denoted in the text as ENGINE<sub>x</sub> where ENGINE<sub>x</sub> can be replaced by either ENGINE1 or ENGINE2.

#### 53.6.1.1 Device Identity Register (DID)

The DID register contains the unique identification information.

Client Select: X

Register index: 0

|        |                            |    |    |    |                            |    |    |    |    |    |    |                            |    |    |    |    |  |
|--------|----------------------------|----|----|----|----------------------------|----|----|----|----|----|----|----------------------------|----|----|----|----|--|
|        | 31                         | 30 | 29 | 28 | 27                         | 26 | 25 | 24 | 23 | 22 | 21 | 20                         | 19 | 18 | 17 | 16 |  |
| R      | Part Revision Number       |    |    |    | Design Center              |    |    |    |    |    |    | Part Identification Number |    |    |    |    |  |
| W      |                            |    |    |    |                            |    |    |    |    |    |    |                            |    |    |    |    |  |
| RESET: | DID_PRN                    |    |    |    | DID_DC                     |    |    |    |    |    |    | DID_PIN                    |    |    |    |    |  |
|        | 15                         | 14 | 13 | 12 | 11                         | 10 | 9  | 8  | 7  | 6  | 5  | 4                          | 3  | 2  | 1  | 0  |  |
| R      | Part Identification Number |    |    |    | Manufacturer Identity Code |    |    |    |    |    |    |                            |    |    |    | 1  |  |
| W      |                            |    |    |    |                            |    |    |    |    |    |    |                            |    |    |    |    |  |
| RESET: | DID_PIN (contd)            |    |    |    | 0                          | 0  | 0  | 0  | 0  | 0  | 0  | 1                          | 1  | 1  | 0  | 1  |  |

**Figure 53-1. Device Identity Register (DID)**

**Table 53-4. DID field descriptions**

| Field  | Description                                                                                                                    |
|--------|--------------------------------------------------------------------------------------------------------------------------------|
| PRN    | Part Revision Number Bits[31:28] contain the revision number of the part.                                                      |
| DC     | Design Center Bits [27:22] indicate the Freescale Design Center.                                                               |
| PIN    | Part Indication Number Bits[21:12] contain the part number of the device.                                                      |
| MIC    | Manufacturer Identity Code Bits [11:1] contain the reduced Joint Electron Device Engineering Council (JEDEC) ID for Freescale. |
| Bit[0] | IDCODE Register ID Bit [0] identifies this register as the device identification register and not the bypass register.         |

#### 53.6.1.2 Client Select Control Register (CSC)

Client Select: X

Register index: 1

Register definition

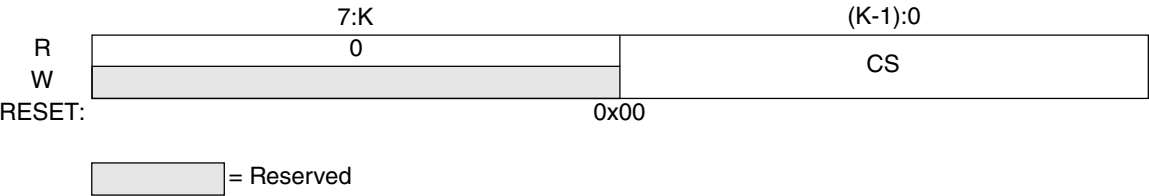


Figure 53-2. Client Select Control Register (CSC)

Table 53-5. CSC field decription

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CS    | <p>Client Select</p> <p>The value of the CS field determines which client is accessed.</p> <p>The number of bits implemented for the CS field (K) is MCU dependent, and is the same number of bits as the size of the SRC field transmitted with messages. Refer to the SoC Guide for the size on a particular MCU.</p> <p>The value written to the CSC register to access a client register is the same value as the SRC field a client uses in its messages. For example, to access an NDEDI ENGINE1 register, the CSC register is written with a value of ENGINE1_SRC.</p> |

53.6.1.3 ENGINE<sub>x</sub> Development Control Register (NDEDI\_ENGINE<sub>x</sub>\_DC)

The NDEDI\_ENGINE<sub>x</sub>\_DC register controls various trace and debug features as described below.

When a field is set to a reserved value the behavior defaults to the reset values behavior.

Client Select: ENGINE<sub>x</sub>\_SRC

Register index: 2

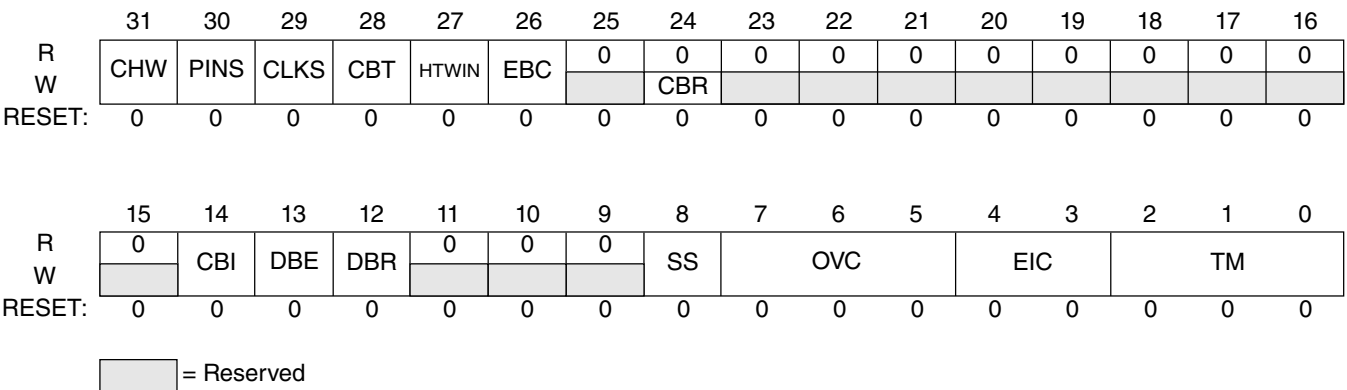


Figure 53-3. ENGINE<sub>x</sub> Development Control Register (NDEDI\_ENGINE<sub>x</sub>\_DC)

**Table 53-6. NDEDI\_ENGINEx\_DC field descriptions**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CHW   | <p>ENGINEx CHAN Register Write Trace</p> <p>The CHW bit enables the tracing of writes to the CHAN register in the ENGINEx execution unit. CHAN register write tracing requires the channel being serviced to have program trace enabled.</p> <p>0 = Tracing not enabled for ENGINEx CHAN writes.</p> <p>1 = Tracing enabled for ENGINEx CHAN writes</p>                                                                                                                                                                                                                                           |
| PINS  | <p>Stop Pins in Debug Mode</p> <p>The PINS bit controls whether the ENGINEx pins are sampled when the ENGINEx enters debug mode. When PINS is set, the pins are not sampled during debug mode or when executing a forced instruction from the microinstruction register. The pins are sampled during single steps.</p> <p>0 = Sample pins in the halted state</p> <p>1 = Stop sampling pins in the halted state</p>                                                                                                                                                                               |
| CLKS  | <p>Stop TCR Clocks</p> <p>The CLKS bit controls whether the TCR clocks from ENGINEx stop running when the ENGINEx enters debug mode. When CLKS is set, TCR is not updated during debug mode or when executing a forced instruction from the microinstruction register. TCR is updated during single steps.</p> <p>0 = Do not stop TCRs during debug mode</p> <p>1 = Stop TCRs during debug mode</p>                                                                                                                                                                                               |
| CBT   | <p>Client Breakpoint Timing</p> <p>The CBT bit controls the timing of an ENGINEx halt due to an external breakpoint source or due to a breakpoint at the twin engine. For this bit to matter, CBI or HTWIN must be set to 1.</p> <p>0 = Halt at the end of the current microcycle</p> <p>1 = Halt at the completion of the current instruction thread</p>                                                                                                                                                                                                                                         |
| HTWIN | <p>HTWIN — Halt on Twin Engine Breakpoint</p> <p>Along with CBT, the HTWIN bit controls the action taken on breakpoints occurrences at the twin engine. If the Twin Engine that is causing a breakpoint exit debug state, the ENGINEx shall resume its operation. If the HTWIN bit is cleared while the ENGINEx is in debug state due to a Twin Engine Breakpoint, the ENGINEx shall resume its operation. This bit must not be cleared when the engine is neither halted nor stopped.</p> <p>0 = Do not halt for breakpoints in the twin Engine</p> <p>1 = Halt for twin Engine's breakpoint</p> |
| EBC   | <p>EVT<math>\overline{O}</math> Breakpoint Controller</p> <p>The EBC bit controls the generation of EVT<math>\overline{O}</math> due to any of the following breakpoint types: Exit of System Reset, DBR assertion, SW, TWIN Engine, EVTI, MISC Error, Illegal Instruction, Single Step or External.</p> <p>0 = Breakpoint status indication is not output on EVT<math>\overline{O}</math></p> <p>1 = Breakpoint status indication is output on EVT<math>\overline{O}</math></p>                                                                                                                  |
| CBR   | <p>Clear Breakpoint Request</p> <p>Writing this bit to one clears breakpoint requests caused by the following conditions:</p> <ul style="list-style-type: none"> <li>• Exit of system reset</li> <li>• Internal breakpoint (due to: instruction address match, accesses to the SPRAM, channel service and writes to the CHAN register)</li> <li>• Single Step breakpoint</li> </ul>                                                                                                                                                                                                               |

*Table continues on the next page...*

**Table 53-6. NDEDI\_ENGINEx\_DC field descriptions (continued)**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | <ul style="list-style-type: none"> <li>• EVTI assertion</li> <li>• Halt instruction at the engine</li> <li>• MISC miscompare</li> <li>• Illegal instruction at the engine.</li> </ul> <p>Asserting this bit causes it to clear one clock later. This means that the development tool can never read this bit with a value other than zero.</p> <p>The ETPU must always execute the first instruction after exiting a breakpoint condition, in such a way that it does not halt twice at the same position.</p> <p>The breakpoints caused by external/system debug request or twin engine are not cleared by the assertion of this bit. This bit must not be asserted when the engine is not halted.</p> <p>0 = No action<br/>1 = Clear breakpoint requests</p>                                                                                                                                                                                                                                                                                                                                                                                            |
| CBI   | <p>Client Breakpoint Input</p> <p>Along with CBT, the CBI bit controls the action taken on breakpoints from external sources.</p> <p>The breakpoint condition caused by the external source is only cleared if CBI equals 0 or if the external breakpoint is negated. This bit must not be cleared when the engine is neither halted nor stopped.</p> <p>0 = Do not halt for other clients' breakpoints<br/>1 = Halt for other clients' breakpoints</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| DBE   | <p>Debug Enable</p> <p>DBE enables debug mode. When DBE is asserted, the occurrence of any of the following events will cause the processor to enter debug mode:</p> <ul style="list-style-type: none"> <li>• DBR assertion</li> <li>• Twin Engine breakpoint</li> <li>• External breakpoint</li> <li>• Exit of system reset</li> <li>• Internal breakpoint (due to: instruction address match, accesses to the SPRAM, channel service and writes to the CHAN register)</li> <li>• Single Step breakpoint</li> <li>• EVTI assertion</li> <li>• Halt instruction at the engine</li> <li>• MISC miscompare</li> <li>• Illegal instruction at the engine.</li> </ul> <p>No breakpoints will occur if DBE is not asserted. If this bit is negated while in debug mode a request to resume the processor operation will be sent to the engine. This bit must not be cleared when the engine is either halted nor stopped.</p> <p>This bit is necessary to use features such as single stepping and breakpoints, but does not affect neither watchpoint, program trace nor data trace operations.</p> <p>0 = Debug mode disabled<br/>1 = Debug mode enabled</p> |

*Table continues on the next page...*

**Table 53-6. NDEDI\_ENGINEx\_DC field descriptions (continued)**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DBR   | <p>Debug Request</p> <p>Mechanism to allow development tool to request processor to enter debug mode. When written, and DBE equals 1 this bit forces the ENGINEx to enter debug mode.</p> <p>The engine will not exit debug mode while this bit is asserted and DBE equals 1.</p> <p>0 = Exit debug mode if there are no pending breakpoint sources</p> <p>1 = Request debug mode</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| OVC   | <p>OVC — Overrun Control</p> <p>The NDEDI can be programmed to stall the ENGINEx operation avoiding errors related to that engine due to full Event Queue. The ENGINEx can't be stalled neither in the middle of an atomic sequence of accesses to the SPRAM nor in the middle of the Time Slot Transition(TST). Please refer to the ETPU BG for further information.</p> <p>The overrun control field controls how the NDEDI reacts when the Queue is about to get full. If programmed to delay the processor, the NDEDI asserts the request to delay the processor when the Event Queue has only STALL_THRESHOLD available positions, and negates the delay request as soon as there are STALL_THRESHOLD + 1 available positions.</p> <p>The request to delay the processor does not affect neither the other engine nor the CDC operations. Therefore, in some extreme cases an overrun message may be generated even with OVC configured to delay the processor. If an error event happens, the processor will be delayed until the Event Queue is enabled for storing snapshots again, which will happen as the Event Queue gets empty.</p> <p>If the OVC field is changed to 0b011 while the Event Queue has less than STALL_THRESHOLD available positions the NDEDI will automatically assert the stall request. If the OVC field is changed to a value other than 0b011 while the ENGINEx is in STALL state, the NDEDI will automatically negate the stall request.</p> <p>A breakpoint request prevails over the request to delay the processor. Thus, if a breakpoint request happens at the same time of a stall request, the ETPU will enter in HALT state instead of STALL state.</p> <p>OVC values:</p> <p>000b = Generate overrun message</p> <p>001b–010b = Reserved</p> <p>011b = Stall Processor to prevent overruns for ENGINEx</p> <p>100b–111b = Reserved</p> |
| EIC   | <p>ENGINEx <math>\overline{\text{EVTI}}</math> Control</p> <p>The <math>\overline{\text{EVTI}}</math> control (EIC) field can be configured for synchronization or breakpoint generation. The <math>\overline{\text{EVTI}}</math> signal is edge sensitive.</p> <p>ENGINEx_EIC values:</p> <p>00b = <math>\overline{\text{EVTI}}</math> assertion causes the next ENGINEx program and data trace message to be a synchronization message (the next CDC data trace messages are synchronized if the next message from either Engines are synchronized).</p> <p>01b = <math>\overline{\text{EVTI}}</math> assertion causes ENGINEx breakpoint generation</p> <p>10b = No operation</p> <p>11b = Reserved</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| TM    | <p>Trace Mode</p> <p>The TM field enables BTM, DTM, and OTM for ENGINEx. One or all types of trace may be enabled at a single time by writing to the TM field.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

**Table 53-6. NDEDI\_ENGINEx\_DC field descriptions**

| Field | Description                                                                                                                                                                                                                                                                                 |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | <p>Note: Data and Program Trace can also be enabled through watchpoint triggers. The two enable sources are ORed together to form the true trace enable.</p> <p>TM values:</p> <p>000b = No trace enabled</p> <p>1xxb = BTM enabled</p> <p>x1xb = DTM enabled</p> <p>xx1b = OTM enabled</p> |

### 53.6.1.4 ENGINEx Development Status Register (NDEDI\_ENGINEx\_DS)

The NDEDI\_ENGINEx\_DS register shows the status of various conditions that impact development support. All status bits are dynamic and do not require clearing. If the Class 3 is enabled, any time the value of any bit in this register changes, a Debug Status Event is queued, and then will generate a Debug Status Message. If the Event Queue is not enabled to store snapshots while a DS event should be queued, then an error event is queued as soon as the Event Queue is enabled for storing snapshots.

DS register dynamically tracks the debug condition status even when several of them are simultaneously asserted. For instance, if EVTI and external debug are asserted at the same time, ETPU will enter in debug mode and NDEDI will assert the bits EBP(EVTI) and CBP. Then, if CBR is asserted the EBP bit will be cleared but the CBP bit will continue to be active since the external debug request is still on.

In some extreme cases NDEDI may not signalize more than one breakpoints condition at the same address. Please refer to [Multiple breakpoint conditions at the same address](#) for more details about this limitation.

Client Select: ENGINEx\_SRC

Register index: 4

|        |    |    |    |     |      |     |     |     |    |    |    |    |    |    |    |    |
|--------|----|----|----|-----|------|-----|-----|-----|----|----|----|----|----|----|----|----|
|        | 31 | 30 | 29 | 28  | 27   | 26  | 25  | 24  | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| R      | 0  | 0  | 0  | HAT | IIBP | CBP | TBP | EBP | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| W      |    |    |    |     |      |     |     |     |    |    |    |    |    |    |    |    |
| RESET: | 0  | 0  | 0  | 0   | 0    | 0   | 0   | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

|        |    |    |    |    |    |    |   |   |   |   |     |     |     |     |     |     |
|--------|----|----|----|----|----|----|---|---|---|---|-----|-----|-----|-----|-----|-----|
|        | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5   | 4   | 3   | 2   | 1   | 0   |
| R      | 0  |    |    |    | BP |    |   |   | 0 | 0 | DBS | STP | HWE | HWB | SWB | SSS |
| W      |    |    |    |    |    |    |   |   |   |   |     |     |     |     |     |     |
| RESET: | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0   | 0   | 0   | 0   | 0   | 0   |

 = Reserved

**Figure 53-4. ENGINEx Development Status Register (NDEDI\_ENGINEx\_DS)****Table 53-7. NDEDI\_ENGINEx\_DS field descriptions**

| Field | Description                                                                                                                                                                                                                                                                                                  |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| HAT   | <p>Halted with Active Thread</p> <p>The HAT bit indicates if the engine was halted during the execution of a thread or while it was IDLE.</p> <p>0 = The ETPU engine is halted with no thread currently active</p> <p>1 = The ETPU engine is halted in the middle of a thread execution</p>                  |
| IIBP  | <p>Illegal Instruction Breakpoint Status</p> <p>The IIBP bit indicates if there is a breakpoint due to an illegal instruction execution causing the halt condition</p> <p>0 = Breakpoint source is not an illegal instruction execution</p> <p>1 = Breakpoint source is an illegal instruction execution</p> |
| CBP   | <p>Client Breakpoint Status</p> <p>The CBP bit indicates if there is a breakpoint from an external client causing the halt condition</p> <p>0 = Breakpoint source is not an external client</p> <p>1 = Breakpoint source is an external client</p>                                                           |
| TBP   | <p>Twin Breakpoint Status</p> <p>The TBP bit indicates if there is a breakpoint from the twin ETPU engine causing the halt condition</p> <p>0 = Breakpoint source is not the twin ETPU engine</p> <p>1 = Breakpoint source is the twin ETPU engine</p>                                                       |
| EBP   | <p>EVTI Breakpoint Status</p> <p>The EBP bit indicates if there is a breakpoint caused by EVTI assertion</p> <p>0 = Breakpoint source is not EVTI assertion</p> <p>1 = Breakpoint source is EVTI assertion</p>                                                                                               |
| BP    | <p>Internal Breakpoint Status</p> <p>The BP field shows which hardware breakpoints<sup>1</sup> have occurred causing the ENGINEx to enter debug mode .</p> <p>BP values:</p> <p>0b00000000 = No Breakpoint Condition Occurred</p> <p>0b1xxxxxx = ETPU Breakpoint 1 (based on BWC1 register)</p>              |

*Table continues on the next page...*

**Table 53-7. NDEDI\_ENGINEx\_DS field descriptions (continued)**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | 0bx1xxxx = ETPU Breakpoint 2 (based on BWC2 register)<br>0bxx1xxxx = Channel Register Write Breakpoint<br>0bxxx1xxx = Host Service Request Breakpoint<br>0bxxxx1xx = Link Register Breakpoint<br>0bxxxxx1x = MRL Breakpoint<br>0bxxxxxx1 = TDL Breakpoint                                                                                                                                                                                                                                                                            |
| DBS   | Debug Status<br>The DBS bit indicates if ENGINEx is in debug mode. The fields HAT, IIBP, CBP, TBP, EBP, BP, HWB, SWB and SSS are only meaningful if the processor is indicated as halted.<br>0 = Processor not halted.<br>1 = Processor halted in debug mode                                                                                                                                                                                                                                                                         |
| STP   | Stop Status<br>Indicates if the ENGINEx is in stop mode . Besides the IPI Green line stop mode mechanism, the ETPU engines may independently be programmed for STOP mode by asserting the MDIS bit of the ECR register. Thus, the STP field indicates if the engine is in stop mode no matter if due to the IPI Green line stop mode assertion or due to the MDIS field. Please refer to the ETPU BG for further information about the stop mode mechanisms.<br>0 = Processor not stopped<br>1 = Processor stopped in low power mode |
| HWE   | Hardware Error Status<br>Indicates if the ENGINEx has encountered a HW error . The only hardware error defined for the ENGINE is a MISC miscompare. Please refer to the ETPU BG for further information about MISC.<br>0 = No HW error<br>1 = Non-recoverable HW error occurred                                                                                                                                                                                                                                                      |
| HWB   | Hardware Breakpoint Status<br>Indicates if one of the hardware breakpoints described at the BP field has occurred .<br>0 = No hardware breakpoint occurred<br>1 = Hardware breakpoint occurred                                                                                                                                                                                                                                                                                                                                       |
| SWB   | Software Breakpoint Status<br>Indicates if a software breakpoint has occurred. A software breakpoint occurs with the execution of the HALT instruction. For further information, please refer to the ETPU BG.<br>0 = No software breakpoint occurred<br>1 = Software breakpoint occurred                                                                                                                                                                                                                                             |
| SSS   | Single Step Status<br>Indicates if the processor is halted for debug mode due to single step occurrence .<br>0 = Processor not halted due to single step<br>1 = Processor halted in debug mode due to single step                                                                                                                                                                                                                                                                                                                    |

1. A hardware breakpoint is a breakpoint configured by BWC1, BWC2 or BWC3.



### 53.6.1.5 ENGINE<sub>x</sub> Watchpoint Trigger Register (NDEDI\_ENGINE<sub>x</sub>\_WT)

The NDEDI\_ENGINE<sub>x</sub>\_WT register allows traces to be enabled and/or disabled on the occurrence of a watchpoint. If the same watchpoint is programmed to enable and disable a trace, the occurrence of the watchpoint toggles the current value of the trace flags. If one watchpoint occurs to enable trace and a different watchpoint occurs to disable trace at the same time the current value of the trace flags will also be toggled. Trace enable flags from triggers (PTEF and DTEF) are ORed with the appropriate bit in the TM field of the DC register and sent to the trace blocks.

Client Select: ENGINE<sub>x</sub>\_SRC

Register index: 11

|        |     |    |    |     |    |    |     |    |    |     |    |    |    |    |    |    |
|--------|-----|----|----|-----|----|----|-----|----|----|-----|----|----|----|----|----|----|
|        | 31  | 30 | 29 | 28  | 27 | 26 | 25  | 24 | 23 | 22  | 21 | 20 | 19 | 18 | 17 | 16 |
| R      | PTS |    |    | PTE |    |    | DTS |    |    | DTE |    |    | 0  | 0  | 0  | 0  |
| W      |     |    |    |     |    |    |     |    |    |     |    |    |    |    |    |    |
| RESET: | 0   | 0  | 0  | 0   | 0  | 0  | 0   | 0  | 0  | 0   | 0  | 0  | 0  | 0  | 0  | 0  |

|        |    |    |    |    |    |    |   |   |   |   |   |   |   |   |      |      |
|--------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|------|------|
|        | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1    | 0    |
| R      | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | PTEF | DTEF |
| W      |    |    |    |    |    |    |   |   |   |   |   |   |   |   | PTEC | DETC |
| RESET: | 0  | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0    | 0    |

 = Reserved

**Figure 53-5. ENGINE<sub>x</sub> Watchpoint Trigger Register (NDEDI\_ENGINE<sub>x</sub>\_WT)**

**Table 53-8. NDEDI\_ENGINE<sub>x</sub>\_WT field descriptions**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PTS   | <p>Program Trace Start</p> <p>This bit allows program trace to be enabled at a watchpoint occurrence.</p> <p>PTS values:</p> <p>0b000 = Trigger Disabled</p> <p>0b001 = Program Trace enabled on ETPU Watchpoint 1 occurrence</p> <p>0b010 = Program Trace enabled on ETPU Watchpoint 2 occurrence</p> <p>0b011 = Program Trace enabled on Channel Register Write Watchpoint occurrence</p> <p>0b100 = Program Trace enabled on Host Service Request Watchpoint occurrence</p> <p>0b101 = Program Trace enabled on Link Register Watchpoint occurrence</p> <p>0b110 = Program Trace enabled on MRL Watchpoint occurrence</p> <p>0b111 = Program Trace enabled on TDL Watchpoint occurrence</p> |
| PTE   | <p>Program Trace End</p> <p>This bit allows program trace to be disabled at a watchpoint occurrence.</p> <p>PTE values:</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

*Table continues on the next page...*

**Table 53-8. NDEDI\_ENGINEx\_WT field descriptions (continued)**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | <p>0b000 = Trigger Disabled</p> <p>0b001 = Program Trace disabled on ETPU Watchpoint 1 occurrence</p> <p>0b010 = Program Trace disabled on ETPU Watchpoint 2 occurrence</p> <p>0b011 = Program Trace disabled on Channel Register Write Watchpoint occurrence</p> <p>0b100 = Program Trace disabled on Host Service Request Watchpoint occurrence</p> <p>0b101 = Program Trace disabled on Link Register Watchpoint occurrence</p> <p>0b110 = Program Trace disabled on MRL Watchpoint occurrence</p> <p>0b111 = Program Trace disabled on TDL Watchpoint occurrence</p>                                                                                                  |
| DTS   | <p>Data Trace Start</p> <p>This bit allows data trace to be enabled at a watchpoint occurrence.</p> <p>DTS values:</p> <p>0b000 = Trigger Disabled</p> <p>0b001 = Data Trace enabled on ETPU Watchpoint 1 occurrence</p> <p>0b010 = Data Trace enabled on ETPU Watchpoint 2 occurrence</p> <p>0b011 = Data Trace enabled on Channel Register Write Watchpoint occurrence</p> <p>0b100 = Data Trace enabled on Host Service Request Watchpoint occurrence</p> <p>0b101 = Data Trace enabled on Link Register Watchpoint occurrence</p> <p>0b110 = Data Trace enabled on MRL Watchpoint occurrence</p> <p>0b111 = Data Trace enabled on TDL Watchpoint occurrence</p>       |
| DTE   | <p>Data Trace End</p> <p>This bit allows data trace to be disabled at a watchpoint occurrence.</p> <p>DTE values:</p> <p>0b000 = Trigger Disabled</p> <p>0b001 = Data Trace disabled on ETPU Watchpoint 1 occurrence</p> <p>0b010 = Data Trace disabled on ETPU Watchpoint 2 occurrence</p> <p>0b011 = Data Trace disabled on Channel Register Write Watchpoint occurrence</p> <p>0b100 = Data Trace disabled on Host Service Request Watchpoint occurrence</p> <p>0b101 = Data Trace disabled on Link Register Watchpoint occurrence</p> <p>0b110 = Data Trace disabled on MRL Watchpoint occurrence</p> <p>0b111 = Data Trace disabled on TDL Watchpoint occurrence</p> |
| PTEF  | <p>Program Trace Enable Flag</p> <p>This bit shows if program trace is currently enabled due to a watchpoint trigger.</p> <p>0 = Program Trace not enabled due to watchpoint trigger</p> <p>1 = Program Trace enabled due to watchpoint trigger</p>                                                                                                                                                                                                                                                                                                                                                                                                                       |
| PTEC  | <p>Program Trace Enable Clear</p> <p>Writing a one to the PTEC bit clears the PTEF flag. This allows the development tool to turn off program trace started via a trigger without using a second trigger or module reset.</p> <p>0 = Keep PTEF flag unaltered</p>                                                                                                                                                                                                                                                                                                                                                                                                         |

*Table continues on the next page...*

**Table 53-8. NDEDI\_ENGINEx\_WT field descriptions (continued)**

| Field | Description                                                                                                                                                                                                                                                             |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | 1 = Clear the PTEF flag                                                                                                                                                                                                                                                 |
| DTEF  | Data Trace Enable Flag<br>This bit shows if data trace is currently enabled due to a watchpoint trigger.<br>0 = Data Trace not enabled due to watchpoint trigger<br>1 = Data Trace enabled due to watchpoint trigger                                                    |
| DTEC  | Data Trace Enable Clear<br>Writing a one to the DTEC bit clears the DTEF flag. This allows the development tool to turn off data trace started via a trigger without using a second trigger or module reset.<br>0 = Keep DTEF flag unaltered<br>1 = Clear the DTEF flag |

### 53.6.1.6 ENGINEx Data Trace Control Register (NDEDI\_ENGINEx\_DTC)

The NDEDI\_ENGINEx\_DTC register controls which address ranges are enabled for data trace, and if reads and/or writes are traced in that range.

Client Select: ENGINEx\_SRC

Register index: 13

|        |      |    |      |    |      |    |      |    |     |     |     |     |    |    |    |    |
|--------|------|----|------|----|------|----|------|----|-----|-----|-----|-----|----|----|----|----|
|        | 31   | 30 | 29   | 28 | 27   | 26 | 25   | 24 | 23  | 22  | 21  | 20  | 19 | 18 | 17 | 16 |
| R      | RWT0 |    | RWT1 |    | RWT2 |    | RWT3 |    | 0   | 0   | 0   | 0   | 0  | 0  | 0  | 0  |
| W      |      |    |      |    |      |    |      |    |     |     |     |     |    |    |    |    |
| RESET: | 0    | 0  | 0    | 0  | 0    | 0  | 0    | 0  | 0   | 0   | 0   | 0   | 0  | 0  | 0  | 0  |
|        | 15   | 14 | 13   | 12 | 11   | 10 | 9    | 8  | 7   | 6   | 5   | 4   | 3  | 2  | 1  | 0  |
| R      | 0    | 0  | 0    | 0  | 0    | 0  | 0    | 0  | RC0 | RC1 | RC2 | RC3 | 0  | 0  | 0  | 0  |
| W      |      |    |      |    |      |    |      |    |     |     |     |     |    |    |    |    |
| RESET: | 0    | 0  | 0    | 0  | 0    | 0  | 0    | 0  | 0   | 0   | 0   | 0   | 0  | 0  | 0  | 0  |

 = Reserved

**Figure 53-6. ENGINEx Data Trace Control Register (NDEDI\_ENGINEx\_DTC)****Table 53-9. NDEDI\_ENGINEx\_DTC field descriptions**

| Field | Description                                                                                                                                                                                                                                                                                                              |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RWT0  | ENGINEx Read/Write Trace 0 Control<br>RWT0 controls whether data trace messages are generated for ENGINEx accesses inside ETPU data trace window 0 (see <a href="#">Data Trace Address Range 0 Register (NDEDI_DTAR0)</a> ), and if so, whether reads, writes, or both generate the data trace messages.<br>RWT0 values: |

*Table continues on the next page...*

**Table 53-9. NDEDI\_ENGINEx\_DTC field descriptions (continued)**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | <p>0b00 = No ENGINEx data trace messages generated for ETPU window 0</p> <p>0b01 = Enable ENGINEx data read trace for ETPU window 0</p> <p>0b10 = Enable ENGINEx data write trace for ETPU window</p> <p>00b11 = Enable ENGINEx data read and write trace for ETPU window 0</p>                                                                                                                                                                                                                                                                                                                                         |
| RWT1  | <p>ENGINEx Read/Write Trace 1 Control</p> <p>RWT1 controls whether data trace messages are generated for ENGINEx accesses inside ETPU data trace window 1 (see <a href="#">Data Trace Address Range 1 Register (NDEDI_DTAR1)</a>), and if so, whether reads, writes, or both generate the data trace messages.</p> <p>RWT1 values:</p> <p>0b00 = No ENGINEx data trace messages generated for ETPU window 1</p> <p>0b01 = Enable ENGINEx data read trace for ETPU window 1</p> <p>0b10 = Enable ENGINEx data write trace for ETPU window 1</p> <p>0b11 = Enable ENGINEx data read and write trace for ETPU window 1</p> |
| RWT2  | <p>ENGINEx Read/Write Trace 2 Control</p> <p>RWT2 controls whether data trace messages are generated for ENGINEx accesses inside ETPU data trace window 2 (see <a href="#">Data Trace Address Range 2 Register (NDEDI_DTAR2)</a>), and if so, whether reads, writes, or both generate the data trace messages.</p> <p>RWT2 values:</p> <p>0b00 = No ENGINEx data trace messages generated for ETPU window 2</p> <p>0b01 = Enable ENGINEx data read trace for ETPU window 2</p> <p>0b10 = Enable ENGINEx data write trace for ETPU window 2</p> <p>0b11 = Enable ENGINEx data read and write trace for ETPU window 2</p> |
| RWT3  | <p>ENGINEx Read/Write Trace 3 Control</p> <p>RWT3 controls whether data trace messages are generated for ENGINEx accesses inside ETPU data trace window 3 (see <a href="#">Data Trace Address Range 3 Register (NDEDI_DTAR3)</a>), and if so, whether reads, writes, or both generate the data trace messages.</p> <p>RWT3 values:</p> <p>0b00 = No ENGINEx data trace messages generated for ETPU window 3</p> <p>0b01 = Enable ENGINEx data read trace for ETPU window 3</p> <p>0b10 = Enable ENGINEx data write trace for ETPU window 3</p> <p>0b11 = Enable ENGINEx data read and write trace for ETPU window 3</p> |
| RC0   | <p>Range Control 0</p> <p>Controls which addresses match data trace window 0 .</p> <p>0 = Trace address inside (inclusive) of data trace window 0</p> <p>1 = Trace addresses outside (exclusive) of data trace window 0</p>                                                                                                                                                                                                                                                                                                                                                                                             |
| RC1   | <p>Range Control 1</p> <p>Controls which addresses match data trace window 1 .</p> <p>0 = Trace address inside (inclusive) of data trace window 1</p> <p>1 = Trace addresses outside (exclusive) of data trace window 1</p>                                                                                                                                                                                                                                                                                                                                                                                             |
| RC2   | Range Control 2                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

*Table continues on the next page...*

**Table 53-9. NDEDI\_ENGINEx\_DTC field descriptions (continued)**

| Field | Description                                                                                                                                                                                              |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | Controls which addresses match data trace window 2 .<br>0 = Trace address inside (inclusive) of data trace window 2<br>1 = Trace addresses outside (exclusive) of data trace window 2                    |
| RC3   | Range Control 3<br>Controls which addresses match data trace window 3 .<br>0 = Trace address inside (inclusive) of data trace window 3<br>1 = Trace addresses outside (exclusive) of data trace window 3 |

### 53.6.1.7 ENGINEx Breakpoint/Watchpoint Control 1/2 Register (NDEDI\_ENGINEx\_BWC1/2)

The ENGINEx Breakpoint/Watchpoint Control 1 Register is used to configure ENGINEx breakpoint/watchpoint 1 along with the NDEDI\_ENGINEx\_BWA1 and NDEDI\_ENGINEx\_BWD1 registers.

The ENGINEx Breakpoint/Watchpoint Control 2 Register is used to configure ENGINEx breakpoint/watchpoint 2 along with the NDEDI\_ENGINEx\_BWA2 and NDEDI\_ENGINEx\_BWD2 registers.

These registers are responsible for configuring the hardware breakpoints/watchpoints 1 and 2. And every field of this register is related to these breakpoints only. When a field is set to a reserved value the behavior defaults to the reset values behavior.

The register can be configured for breakpoint/watchpoint based on data write values, instruction address and data read or write addresses.

When a hardware breakpoint occurs the program execution is halted before the instruction is executed. This means that data access breakpoints occur before the SPRAM access actually takes place.

The data accesses that occur within the Time Slot Transition (TST) do not generate a hardware breakpoint. Thus, it is not expected that the ETPU system signalize these transactions to the NDEDI interface.

Client Select: ENGINEx\_SRC

Register index: 22

## Register definition

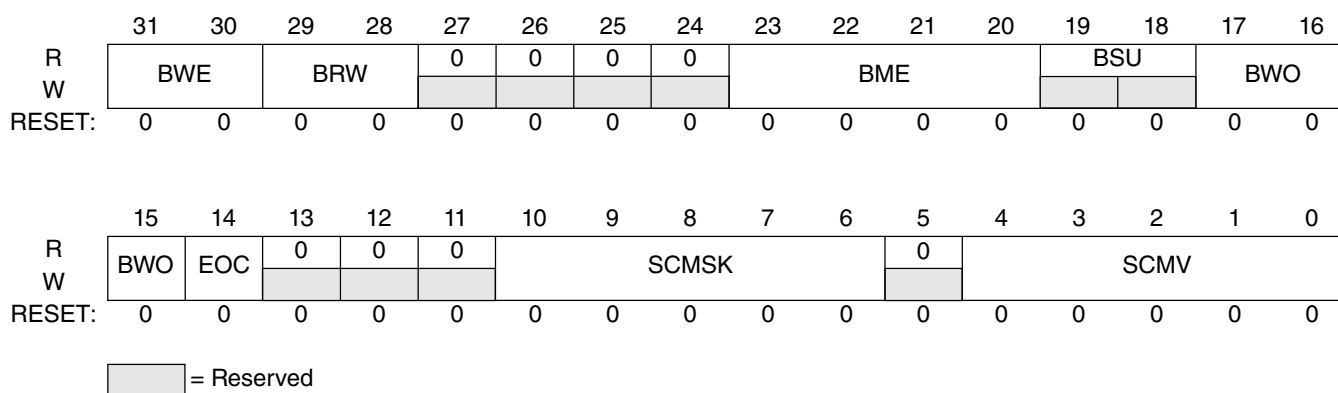


**Figure 53-7. ENGINE<sub>x</sub> Breakpoint/Watchpoint Control 1 Register (NDEDI\_ENGINE<sub>x</sub>\_BWC1)**

Client Select: ENGINE<sub>x</sub>\_SRC

Register index: 23

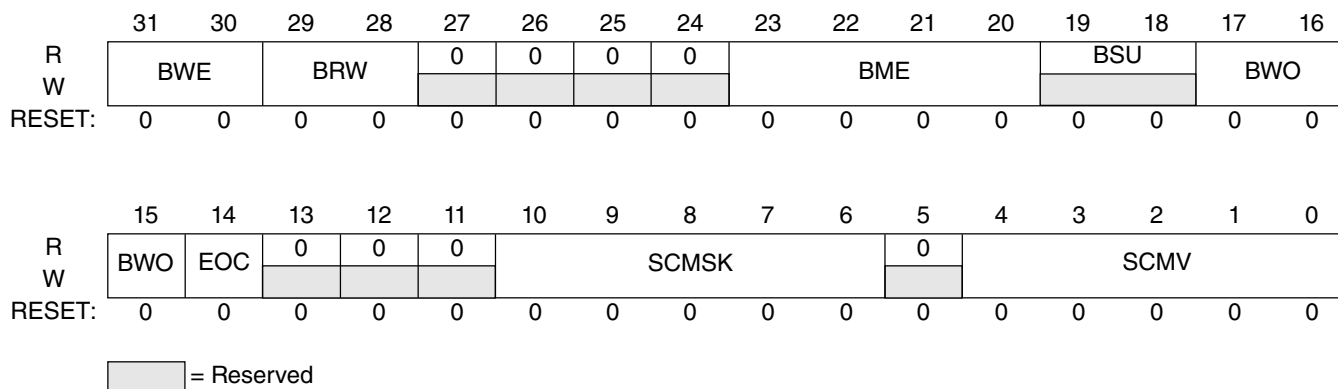


**Figure 53-8. ENGINE<sub>x</sub> Breakpoint/Watchpoint Control 2 Register (NDEDI\_ENGINE<sub>x</sub>\_BWC2)**

**Table 53-10. NDEDI\_ENGINE<sub>x</sub>\_BWC1/2 field descriptions**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| BWE   | <p>Breakpoint/Watchpoint Enable</p> <p>The BWE field is used to enable the ENGINE<sub>x</sub> hardware breakpoint or watchpoint.</p> <p>BWE values:</p> <p>0b00 = Disabled</p> <p>0b01 = Breakpoint Enabled</p> <p>0b10 = Reserved</p> <p>0b11 = Watchpoint Enabled</p> <p>Changing BWE from 'Breakpoint Enabled'(BWE=0b01) to 'Watchpoint Enabled'(BWE=0b11) and vice-versa must only be done while ETPU is either halted or stopped.</p> <p>To set BWE as either 'Breakpoint Enabled'(BWE=0b01) or 'Watchpoint Enabled'(BWE=0b11) while ETPU is neither halted nor stopped the user must previously set BWE as 'Disabled' (BWE=0b00).</p> <p>All other fieds of this register must only be changed when BWE=0b00 (disabled) when the engine is not stopped or halted, in 3 steps:</p> |

*Table continues on the next page...*

**Table 53-10. NDEDI\_ENGINEx\_BWC1/2 field descriptions (continued)**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | (a) write BWE=0b00; (b) change other fields; (c) write BWE with a valid non-zero value, keeping the other fields unchanged.                                                                                                                                                                                                                                                                                                                                                                                                                      |
| BRW   | <p>Breakpoint/Watchpoint Read/Write Select</p> <p>The BRW field selects if read and/or write accesses cause a hardware breakpoint/watchpoint.</p> <p>BRW values:</p> <p>0b00 = Match on read SPRAM access</p> <p>0b01 = Match on write SPRAM access</p> <p>0b10 = Match on any SPRAM accesses</p> <p>0b11 = Reserved</p>                                                                                                                                                                                                                         |
| BME   | <p>Breakpoint/Watchpoint Data Mask Enable</p> <p>The BME field selects which data bytes are used for hardware breakpoint/watchpoint generation.</p> <p>BME values:</p> <p>0b0000 = All bytes of data compared</p> <p>0b1xxx = Most significant byte of data masked out</p> <p>...</p> <p>0bxxx1 = Least significant byte of data masked out</p>                                                                                                                                                                                                  |
| BSU   | <p>Breakpoint/Watchpoint Data Size Unit</p> <p>The BSU is a read-only field to indicate the data size unit used by the NDEDI block. This field always reads as 0b00 to indicate the data size unit is 1 byte.</p>                                                                                                                                                                                                                                                                                                                                |
| BWO   | <p>Breakpoint/Watchpoint Operand</p> <p>The BWO selects if address and/or data matching is done and if matching is done on data fetches or instruction fetches.</p> <p>BWO values:</p> <p>0b00x = Breakpoint/Watchpoint Disabled</p> <p>0bx10 = Reserved</p> <p>0b011 = Compare data of a SPRAM write access with BWD</p> <p>0b100 = Compare instruction Address with BWA</p> <p>0b101 = Compare Address of a SPRAM write/read access with BWA</p> <p>0b111 = Compare Data and Address of a SPRAM write access with BWD and BWA respectively</p> |
| EOC   | <p><math>\overline{\text{EVT0}}</math> Control</p> <p>The EOC bit controls the action of the <math>\overline{\text{EVT0}}</math> output at the occurrence of a hardware breakpoint or watchpoint.</p> <p>0 = Breakpoint/Watchpoint status indication is not output on <math>\overline{\text{EVT0}}</math></p> <p>1 = Breakpoint/Watchpoint status indication is output on <math>\overline{\text{EVT0}}</math></p>                                                                                                                                |
| SCMSK | <p>Serviced Channel Mask Value</p> <p>The SCMSK field is used to mask what bits of the SCMV field are compared to the serviced channel. When a bit in the SCMSK field is 0, that bit is not compared for masking purposes.</p> <p>serviced channel match = (SCMV&amp;SCMSK) == (Serviced Channel &amp; SCMSK);</p>                                                                                                                                                                                                                               |
| SCMV  | Serviced Channel Match Value                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

Table 53-10. NDEDI\_ENGINEx\_BWC1/2 field descriptions

| Field | Description                                                                                                            |
|-------|------------------------------------------------------------------------------------------------------------------------|
|       | The SCMV field value is compared against the ENGINEx serviced channel when generating the breakpoints and watchpoints. |

53.6.1.8 ENGINEx Breakpoint/Watchpoint Address 1/2 Register (NDEDI\_ENGINEx\_BWA1/2)

The ENGINEx Breakpoint/Watchpoint Address 1 Register is used to configure ENGINEx hardware breakpoint/watchpoint 1 along with the NDEDI\_ENGINEx\_BWC1 and NDEDI\_ENGINEx\_BWD1 registers. It is used for data access and instruction fetches breakpoints.

The ENGINEx Breakpoint/Watchpoint Address 2 Register is used to configure ENGINEx hardware breakpoint/watchpoint 2 along with the NDEDI\_ENGINEx\_BWC2 and NDEDI\_ENGINEx\_BWD2 registers. It is used for data access and instruction fetches breakpoints.

Client Select: ENGINEx\_SRC

Register index: 30

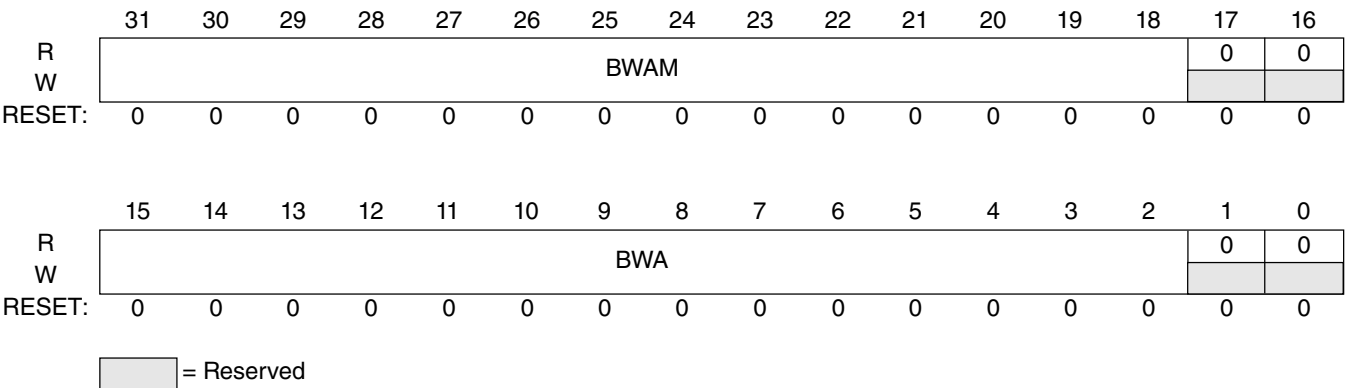


Figure 53-9. ENGINEx Breakpoint/Watchpoint Address 1 Register (NDEDI\_ENGINEx\_BWA1)

Client Select: ENGINEx\_SRC

Register index: 31



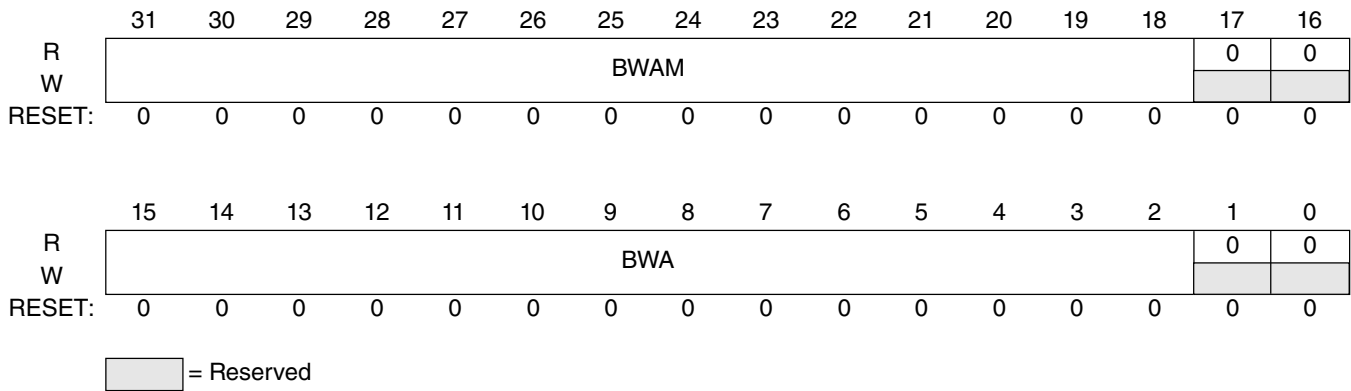


**Figure 53-10. ENGINEx Breakpoint/Watchpoint Address 2 Register (NDEDI\_ENGINEx\_BWA2)**

**Table 53-11. NDEDI\_ENGINEx\_BWA1/2 field descriptions**

| Field | Description                                                                                                                                                                                                                                                                                               |
|-------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| BWAM  | Breakpoint/Watchpoint Address Mask<br>The BWAM field is used to select which address bits are compared for breakpoint and watchpoint generation.<br>address match = (BWA&BWAM) == (Address & BWAM);                                                                                                       |
| BWA   | Breakpoint/Watchpoint Address<br>The BWA field is used to compare address operands (address of instruction or data).<br>The BWA field represent a word addressable address.<br>The most significant two bits are ignored for address of data comparisons because the SPRAM address space is only 14 bits. |

### 53.6.1.9 ENGINEx Breakpoint/Watchpoint Data 1/2 Register (NDEDI\_ENGINEx\_BWD1/2)

The ENGINEx Breakpoint/Watchpoint Data 1 Register is used to configure the ENGINEx hardware breakpoint/watchpoint 1 along with the NDEDI\_ENGINEx\_BWC1 and NDEDI\_ENGINEx\_BWA1 registers. It is used for data access breakpoints only.

The ENGINEx Breakpoint/Watchpoint Data 2 Register is used to configure the ENGINEx hardware breakpoint/watchpoint 2 along with the NDEDI\_ENGINEx\_BWC2 and NDEDI\_ENGINEx\_BWA2 registers. It is used for data access breakpoints only.

Client Select: ENGINEx\_SRC

Register index: 38

Register definition

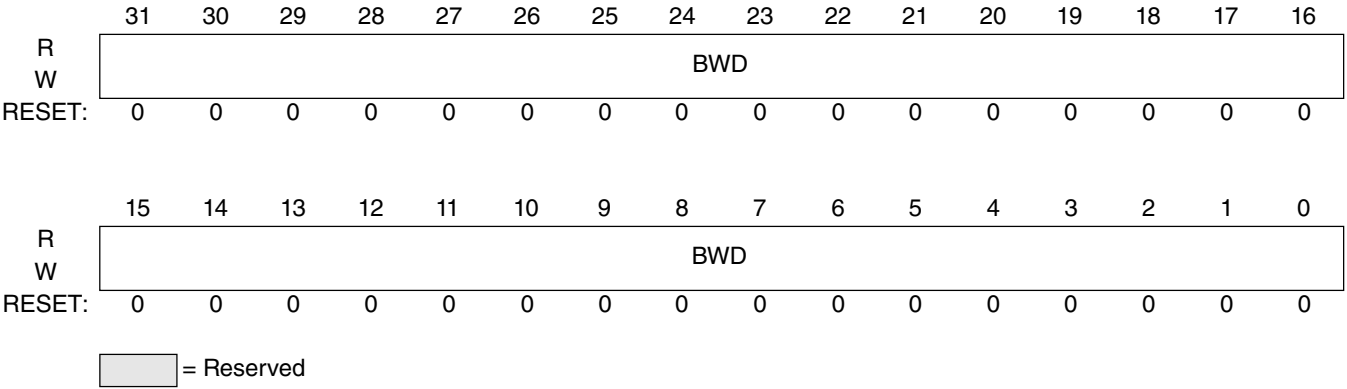


Figure 53-11. ENGINEx Breakpoint/Watchpoint Data 1 Register (NDEDI\_ENGINEx\_BWD1)

Client Select: ENGINEx\_SRC

Register index: 39

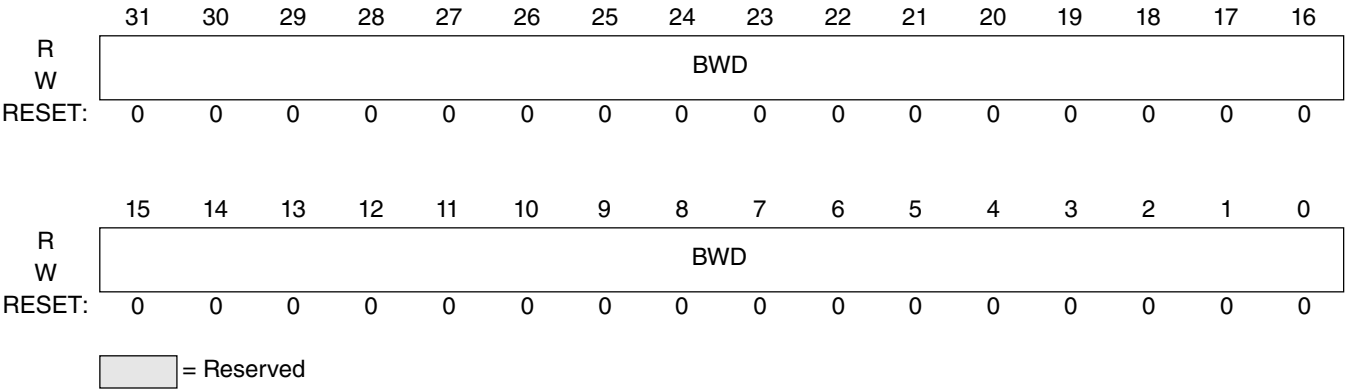


Figure 53-12. ENGINEx Breakpoint/Watchpoint Data 2 Register (NDEDI\_ENGINEx\_BWD2)

Table 53-12. NDEDI\_ENGINEx\_BWD1/2 field descriptions

| Field | Description                                                                                                                                                                                                                                                                                                                                 |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| BWD   | Breakpoint/Watchpoint Data<br><br>The BWD field is used to compare data operands whether in a SPRAM write access. Depending on the size of the access, only certain bits of BWD are compared against the data written to the SPRAM. <a href="#">Table 53-13</a> shows which bits of BWD are compared against the data written to the SPRAM. |

Table 53-13. Bits compared against BWD

| Operation size | Bits compared                    |
|----------------|----------------------------------|
| 8-bits         | 8 most significant bits of BWD   |
| 24-bits        | 24 least significant bits of BWD |
| 32-bits        | All bits from BWD                |

### 53.6.1.10 ENGINEx Program Trace Channel Enable Register (NDEDI\_ENGINEx\_PTCE)

The NDEDI\_ENGINEx\_PTCE register enables program tracing for each of the 32 ENGINEx channels. There is one enable bit per channel. And the bit corresponding to the serviced channel is the only bit tested for enabling program trace. For these bits to have any effect, program trace must also be enabled in the TM field of the DC register or via a watchpoint trigger.

Client Select: ENGINEx\_SRC

Register index: 64

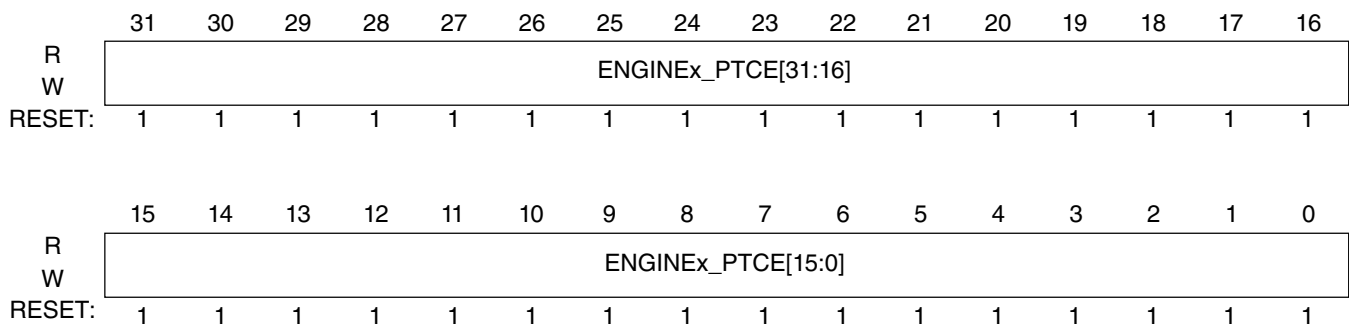


**Figure 53-13. ENGINEx Program Trace Channel Enable Register (NDEDI\_ENGINEx\_PTCE)**

**Table 53-14. NDEDI\_ENGINEx\_PTCE field descriptions**

| Field         | Description                                                                                                                                                         |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ENGINEx_PTCEy | ENGINEx Timer Channel y Program Trace Enable<br>0 = Program tracing disabled for ENGINEx timer channel y<br>1 = Program tracing enabled for ENGINEx timer channel y |

### 53.6.1.11 ENGINEx Breakpoint/Watchpoint Control 3 Register (NDEDI\_ENGINEx\_BWC3)

The ENGINEx Breakpoint/Watchpoint Control 3 Register is used to configure miscellaneous ENGINEx hardware breakpoint/watchpoint sources. This is a vendor-defined register with different fields than BWC1 and BWC2.

When a field is set to a reserved value the behavior defaults to the reset values behavior.

Client Select: ENGINEx\_SRC

## Register definition

Register index: 24

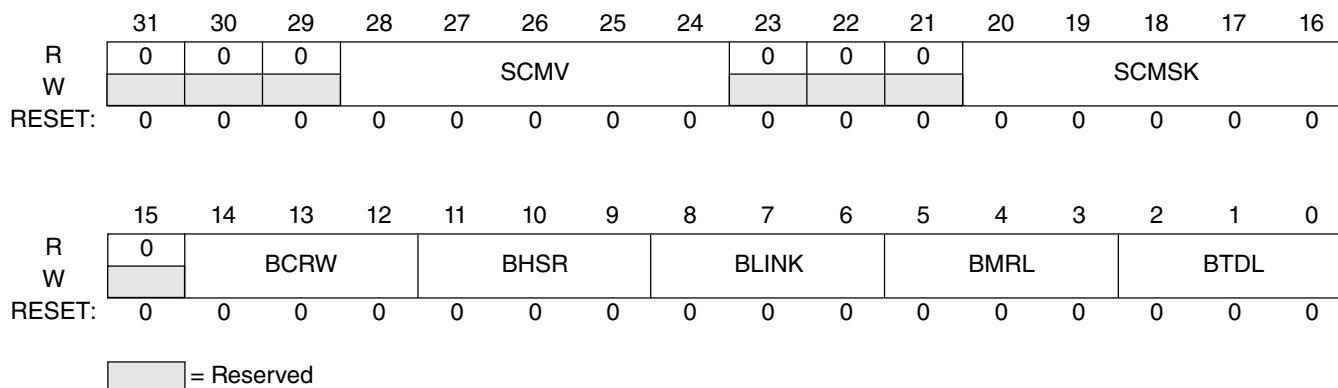


**Figure 53-14. ENGINE<sub>x</sub> Breakpoint/Watchpoint Control 3 Register (NDEDI\_ENGINE<sub>x</sub>\_BWC3)**

**Table 53-15. NDEDI\_ENGINE<sub>x</sub>\_BWC3 field descriptions**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SCMV  | <p>Serviced Channel Match Value</p> <p>The SCMV field value is compared against the serviced channel register when generating a breakpoint due to a write in the CHAN register or due to the start of a service.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| SCMSK | <p>Serviced Channel Mask Value</p> <p>The CMSK field is used to mask what bits of the SCMV field are compared to the serviced channel register when generating a breakpoint due to a write in the CHAN register or due to the start of a service. When a bit in the SCMSK field is 0, that bit is not compared for masking purposes.</p>                                                                                                                                                                                                                                                                                                                                                                         |
| BCRW  | <p>Break on Channel Register Write</p> <p>The BCRW field configures the action when there is a write to the CHAN register and there is a serviced channel match (as described below). The breakpoint causes execution to halt before the CHAN register is written.</p> <p>serviced channel match = (SCMV&amp;SCMSK) == (Serviced Channel &amp; SCMSK);</p> <p>BCRW values:</p> <p>0b00x = Disabled</p> <p>0b01x = Breakpoint Enabled</p> <p>0b10x = Reserved</p> <p>0b11x = Watchpoint Enabled</p> <p>0bxx0 = Breakpoint/Watchpoint status indication not output on <math>\overline{\text{EVT0}}</math></p> <p>0bxx1 = Breakpoint/Watchpoint status indication output on <math>\overline{\text{EVT0}}</math></p> |
| BHSR  | <p>Break on Host Service Request</p> <p>The BHSR field configures the action when there is a serviced channel match (as described below) and the host service request is asserted at the beginning of the channel service. The breakpoint causes execution to halt after the time slot transition (TST) completes but before the first instruction of the thread is executed.</p> <p>serviced channel match = (SCMV&amp;SCMSK) == (Serviced Channel &amp; SCMSK);</p> <p>BHSR values:</p> <p>0b00x = Disabled</p>                                                                                                                                                                                                |

*Table continues on the next page...*

**Table 53-15. NDEDI\_ENGINEx\_BWC3 field descriptions (continued)**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | 0b01x = Breakpoint Enabled<br>0b10x = Reserved<br>0b11x = Watchpoint Enabled<br>0bxx0 = Breakpoint/Watchpoint status indication not output on $\overline{EVTO}$<br>0bxx1 = Breakpoint/Watchpoint status indication output on $\overline{EVTO}$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| BLINK | Break on Link Service Request<br><p>The BLINK field configures the action when there is a serviced channel match (as described bellow) and the link service register is asserted at the beginning of the channel service. The breakpoint causes execution to halt after the time slot transition (TST) completes but before the first instruction of the thread is executed.</p> <p>serviced channel match = (SCMV&amp;SCMSK) == (Serviced Channel &amp; SCMSK);</p> <p>BLINK values:</p> 0b00x = Disabled<br>0b01x = Breakpoint Enabled<br>0b10x = Reserved<br>0b11x = Watchpoint Enabled<br>0bxx0 = Breakpoint/Watchpoint status indication not output on $\overline{EVTO}$<br>0bxx1 = Breakpoint/Watchpoint status indication output on $\overline{EVTO}$ |
| BMRL  | Break on Match Recognition Request<br><p>The BMRL field configures the action when there is a serviced channel match (as described bellow) and either MRL1 or MRL2 is asserted at the beginning of the channel service. The breakpoint causes execution to halt after the time slot transition (TST) completes but before the first instruction of the thread is executed.</p> <p>serviced channel match = (SCMV&amp;SCMSK) == (Serviced Channel &amp; SCMSK);</p> <p>BMRL values:</p> 0b00x = Disabled<br>0b01x = Breakpoint Enabled<br>0b10x = Reserved 1<br>0b11x = Watchpoint Enabled<br>0bxx0 = Breakpoint/Watchpoint status indication not output on $\overline{EVTO}$<br>0bxx1 = Breakpoint/Watchpoint status indication output on $\overline{EVTO}$  |
| BTDL  | Break on Transition Detect Request<br><p>The BTDL field configures the action when there is a serviced channel match (as described bellow) and the TDL1 or TDL2 is asserted at the beginning of the channel service. The breakpoint causes execution to halt after the time slot transition (TST) completes but before the first instruction of the thread is executed.</p> <p>serviced channel match = (SCMV&amp;SCMSK) == (Serviced Channel &amp; SCMSK);</p> <p>BTDL values:</p> 0b00x = Disabled<br>0b01x = Breakpoint Enabled                                                                                                                                                                                                                           |

**Table 53-15. NDEDI\_ENGINEx\_BWC3 field descriptions**

| Field | Description                                                                            |
|-------|----------------------------------------------------------------------------------------|
|       | 0b10x = Reserved                                                                       |
|       | 0b11x = Watchpoint Enabled                                                             |
|       | 0bxx0 = Breakpoint/Watchpoint status indication not output on $\overline{\text{EVTO}}$ |
|       | 0bxx1 = Breakpoint/Watchpoint status indication output on $\overline{\text{EVTO}}$     |

### 53.6.1.12 ENGINEx Microinstruction Debug Register (NDEDI\_ENGINEx\_INST)

The NDEDI\_ENGINEx\_INST register is used for forcing a microinstruction on the microinstruction register. This register must only be accessed while the ENGINEx is halted. Writes to this register are ignored if accessed while the ENGINEx is not halted.

Client Select: ENGINEx\_SRC

Register index: 69

|        |              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|--------|--------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
|        | 31           | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| R      | 0            | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| W      | ENGINEx_INST |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| RESET: | 0            | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

|        |              |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
|        | 15           | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| R      | 0            | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| W      | ENGINEx_INST |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| RESET: | 0            | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 53-15. ENGINEx Microinstruction Debug Register (NDEDI\_ENGINEx\_INST)**

Writes to the register in debug mode force the execution of an arbitrary instruction. This is used to retrieve information about internal status, set internal flag values among other things during debug mode. Some fields of this microinstruction may be ignored, for instance an END instruction will be ignored. For further information about the arbitrary instruction refer to the ETPU BG Section 4.10.2.7.

Reads from the register returns 0.

**Table 53-16. NDEDI\_ENGINEx\_INST field descriptions**

| Field         | Description                                                    |
|---------------|----------------------------------------------------------------|
| ENGINEx_INSTT | the microinstruction forced into the microinstruction register |

### 53.6.1.13 ENGINE<sub>x</sub> Microprogram Counter Debug Register (NDEDI\_ENGINE<sub>x</sub>\_MPC)

The NDEDI\_ENGINE<sub>x</sub>\_MPC register is used for reading the microprogram counter value. This register is only meaningful while the ENGINE<sub>x</sub> is halted. If accessed while the ENGINE<sub>x</sub> is not halted, reads return zero. Writes to this register are always ignored.

Client Select: ENGINE<sub>x</sub>\_SRC

Register index: 70

|        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|--------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
|        | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| R      | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
| W      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| RESET: | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

|        |                                |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|--------|--------------------------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
|        | 15                             | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| R      | NDEDI_ENGINE <sub>x</sub> _MPC |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| W      |                                |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| RESET: | 0                              | 0  | 0  | 0  | 0  | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Figure 53-16. ENGINE<sub>x</sub> Microprogram Counter Debug Register (NDEDI\_ENGINE<sub>x</sub>\_MPC)**

**Table 53-17. NDEDI\_ENGINE<sub>x</sub>\_MPC field descriptions**

| Field                          | Description                |
|--------------------------------|----------------------------|
| NDEDI_ENGINE <sub>x</sub> _MPC | Microprogram counter value |

### 53.6.1.14 ENGINE<sub>x</sub> Channel Flag Status Register (NDEDI\_ENGINE<sub>x</sub>\_CFSR)

The NDEDI\_ENGINE<sub>x</sub>\_CFSR register provides read-only access to the flags of the channel being serviced by ENGINE<sub>x</sub>. Reads to this register when ENGINE<sub>x</sub> is out of halt mode return zero., Transition Detection and Match Recognition flags are available in two versions: One reflecting the Branch Condition Selection (BCC), TDL1/2 and MRL1/2, and an ‘internal’ version, ITDL1/2 and IMR1/2, reflecting the flag value within the channel logic.

Client Select: ENGINE<sub>x</sub>\_SRC

Register index: 71

## Register definition

|        |    |    |    |    |    |    |    |    |      |      |      |      |     |        |    |    |
|--------|----|----|----|----|----|----|----|----|------|------|------|------|-----|--------|----|----|
|        | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23   | 22   | 21   | 20   | 19  | 18     | 17 | 16 |
| R      | V  | N  | C  | Z  | MV | MN | MC | MZ | TDL1 | TDL2 | MRL1 | MRL2 | LSR | SML-CK | FM |    |
| W      |    |    |    |    |    |    |    |    |      |      |      |      |     |        |    |    |
| RESET: | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0    | 0    | 0    | 0   | 0      | 0  | 0  |

|        |     |     |    |    |        |        |        |        |       |       |        |        |     |      |      |     |
|--------|-----|-----|----|----|--------|--------|--------|--------|-------|-------|--------|--------|-----|------|------|-----|
|        | 15  | 14  | 13 | 12 | 11     | 10     | 9      | 8      | 7     | 6     | 5      | 4      | 3   | 2    | 1    | 0   |
| R      | SRI | HSR |    |    | FLAG 0 | FLAG 1 | MRLE 1 | MRLE 2 | ITDL1 | ITDL2 | IMRL 1 | IMRL 2 | PSS | PSTI | PSTO | OBE |
| W      |     |     |    |    |        |        |        |        |       |       |        |        |     |      |      |     |
| RESET: | 0   | 0   | 0  | 0  | 0      | 0      | 0      | 0      | 0     | 0     | 0      | 0      | 0   | 0    | 0    | 0   |

**Figure 53-17. ENGINE<sub>x</sub> Channel Flag Status Register (NDEDI\_ENGINE<sub>x</sub>\_CSFR)**

**Table 53-18. NDEDI\_ENGINE<sub>x</sub>\_CFSR field descriptions**

| Field | Description                                                                                                                                                                          |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| V     | EAU Overflow Flag<br>Overflow status flag from an EAU operation.<br>0 = EAU overflow flag is cleared<br>1 = EAU overflow flag is set                                                 |
| N     | EAU Negative Flag<br>Negative status flag from an EAU operation.<br>0 = EAU negative flag is cleared<br>1 = EAU negative flag is set                                                 |
| C     | EAU Carry Flag<br>Carry status flag from an EAU operation.<br>0 = EAU carry flag is cleared<br>1 = EAU carry flag is set                                                             |
| Z     | EAU Zero Flag<br>Zero status flag from an EAU operation.<br>0 = EAU zero flag is cleared<br>1 = EAU zero flag is set                                                                 |
| MV    | MAC/Divide Unit Overflow Flag<br>Overflow status flag from an MAC/Divide Unit operation.<br>0 = MAC/Divide Unit overflow flag is cleared<br>1 = MAC/Divide Unit overflow flag is set |
| MN    | MAC/Divide Unit Negative Flag<br>Negative status flag from an MAC/Divide Unit operation.<br>0 = MAC/Divide Unit negative flag is cleared<br>1 = MAC/Divide Unit negative flag is set |
| MC    | MAC/Divide Unit Carry Flag<br>Carry status flag from an MAC/Divide Unit operation.<br>0 = MAC/Divide Unit carry flag is cleared                                                      |

*Table continues on the next page...*



**Table 53-18. NDEDI\_ENGINEx\_CFSR field descriptions  
(continued)**

| Field | Description                                                                                                                                                                                                   |
|-------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | 1 = MAC/Divide Unit carry flag is set                                                                                                                                                                         |
| MZ    | MAC/Divide Unit Carry Flag<br>Zero status flag from an MAC/Divide Unit operation.<br>0 = MAC/Divide Unit zero flag is cleared<br>1 = MAC/Divide Unit zero flag is set                                         |
| TDL1  | Channel Transition Detection Latch 1<br>Channel Transition Detection status flag 1 as available in the BCC.<br>0 = Channel Transition flag 1 in BCC is cleared<br>1 = Channel Transition flag 1 in BCC is set |
| TDL2  | Channel Transition Detection Latch 2<br>Channel Transition Detection status flag 2 as available in the BCC.<br>0 = Channel Transition flag 2 in BCC is cleared<br>1 = Channel Transition flag 2 in BCC is set |
| MRL1  | Channel Match Recognition Latch 1<br>Channel Match Recognition status flag 1 as available in the BCC.<br>0 = Match Recognition flag 1 in BCC is cleared<br>1 = Match Recognition flag 1 in BCC is set         |
| MRL2  | Channel Match Recognition Latch 2<br>Channel Match Recognition status flag 2 as available in the BCC.<br>0 = Match Recognition flag 2 in BCC is cleared<br>1 = Match Recognition flag 2 in BCC is set         |
| LSR   | Channel Link Service Register<br>Channel Link Service Register flag.<br>0 = Link even to currently serviced channel did not occur<br>1 = Link event to currently serviced channel occurred                    |
| SMLCK | Channel Semaphore Flag<br>Channel Semaphore flag.<br>0 = Semaphore flag is cleared<br>1 = Semaphore flag is set                                                                                               |
| FM    | Channel Function Mode<br>Reflects the state of the FM field of the Status Control Register of the current channel.                                                                                            |
| SRI   | Service Request Inhibit<br>Flag that blocks channel service requests due to the assertion of MRL1/2 and/or TDL1/2.<br>0 = Service requests not inhibited<br>1 = Service requests inhibited                    |
| HSR   | Host Service Request                                                                                                                                                                                          |

*Table continues on the next page...*

**Table 53-18. NDEDI\_ENGINEx\_CFSR field descriptions  
(continued)**

| Field | Description                                                                                                                                                                                                                                                                                    |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | Indicates the Entry Point HSR of the thread currently executing, if any. This was the value of HSR of the serviced channel when the entry point was chosen. It is not necessarily the value of the Host Service Request Register of the current channel.                                       |
| FLAG0 | Channel State Resolution Flag 0<br><br>One of two flags per channel that can be set or cleared by microcode to further resolve the entry point for a channel service.<br><br>0 = FLAG0 was cleared by microcode<br>1 = FLAG0 was set by microcode                                              |
| FLAG1 | Channel State Resolution Flag 1<br><br>One of two flags per channel that can be set or cleared by microcode to further resolve the entry point for a channel service.<br><br>0 = FLAG1 was cleared by microcode<br>1 = FLAG1 was set by microcode                                              |
| MRLE1 | Channel Match Recognition Latch Enable 1<br>Channel Match Recognition Latch Enable 1.<br><br>0 = Match recognition disabled for event 1<br>1 = Match recognition enabled for event 1                                                                                                           |
| MRLE2 | Channel Match Recognition Latch Enable 2<br>Channel Match Recognition Latch Enable 2.<br><br>0 = Match recognition disabled for event 2<br>1 = Match recognition enabled for event 2                                                                                                           |
| ITDL1 | Internal Channel Transition Detection Latch 1<br><br>Channel Transition Detection status flag 1 reflecting its status directly in the channel logic.<br><br>0 = Channel Transition flag 1 within the channel logic is cleared<br>1 = Channel Transition flag 1 within the channel logic is set |
| ITDL2 | Internal Channel Transition Detection Latch 2<br><br>Channel Transition Detection status flag 2 reflecting its status directly in the channel logic.<br><br>0 = Channel Transition flag 2 within the channel logic is cleared<br>1 = Channel Transition flag 2 within the channel logic is set |
| IMRL1 | Internal Channel Match Recognition Latch 1<br><br>Channel Match Recognition status flag 1 reflecting its status directly in the channel logic.<br><br>0 = Match Recognition flag 1 within the channel logic is cleared<br>1 = Match Recognition flag 1 within the channel logic is set         |
| IMRL2 | Internal Channel Match Recognition Latch 2<br><br>Channel Match Recognition status flag 2 reflecting its status directly in the channel logic.<br><br>0 = Match Recognition flag 2 within the channel logic is cleared<br>1 = Match Recognition flag 2 within the channel logic is set         |
| PSS   | Pin Sampled State                                                                                                                                                                                                                                                                              |

*Table continues on the next page...*

**Table 53-18. NDEDI\_ENGINEx\_CFSR field descriptions  
(continued)**

| Field | Description                                                                                                                                                                                      |
|-------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | Pin Sampled State reflects the status of the PSS flag for the current channel.<br>0 = PSS is cleared<br>1 = PSS is set                                                                           |
| PSTI  | Pin State Input<br>Pin State Input reflects the status of the filtered input signal for the current channel.<br>0 = Filtered input signal is cleared<br>1 = Filtered input signal is set         |
| PSTO  | Pin State Output reflects the status of the value driven by the output control logic of the current channel.<br>0 = Output pin state register is cleared<br>1 = Output pin state register is set |
| OBE   | Output Buffer Enable<br>Output Buffer Enable reflects the status of the OBE control register for the current channel.<br>0 = Output Buffer Enable is cleared<br>1 = Output Buffer Enable is set  |

**53.6.1.15 CDC Data Trace Control Register (NDEDI\_CDC\_DTC)**

The NDEDI\_CDC\_DTC register controls which address ranges are enabled for CDC access data trace, and if reads and/or writes are traced in that range.

Client Select: CDC\_SRC

Register index: 13

|        |      |    |      |    |      |    |      |    |     |    |     |     |     |    |    |    |
|--------|------|----|------|----|------|----|------|----|-----|----|-----|-----|-----|----|----|----|
|        | 31   | 30 | 29   | 28 | 27   | 26 | 25   | 24 | 23  | 22 | 21  | 20  | 19  | 18 | 17 | 16 |
| R      | RWT0 |    | RWT1 |    | RWT2 |    | RWT3 |    | 0   | 0  | 0   | 0   | 0   | 0  | 0  | 0  |
| W      |      |    |      |    |      |    |      |    |     |    |     |     |     |    |    |    |
| RESET: | 0    | 0  | 0    | 0  | 0    | 0  | 0    | 0  | 0   | 0  | 0   | 0   | 0   | 0  | 0  | 0  |
|        | 15   | 14 | 13   | 12 | 11   | 10 | 9    | 8  | 7   | 6  | 5   | 4   | 3   | 2  | 1  | 0  |
| R      | 0    | 0  | 0    | 0  | 0    | 0  | 0    | 0  | RC0 |    | RC1 | RC2 | RC3 | 0  | 0  | 0  |
| W      |      |    |      |    |      |    |      |    |     |    |     |     |     |    |    |    |
| RESET: | 0    | 0  | 0    | 0  | 0    | 0  | 0    | 0  | 0   | 0  | 0   | 0   | 0   | 0  | 0  | 0  |

 = Reserved

**Figure 53-18. CDC Data Trace Control Register (NDEDI\_CDC\_DTC)**

**Table 53-19. NDEDI\_CDC\_DTC field description**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RWT0  | <p>CDC Read/Write Trace 0 Control</p> <p>RWT0 controls whether data trace messages are generated for CDC accesses inside ETPU data trace window 0 (see <a href="#">Data Trace Address Range 0 Register (NDEDI_DTAR0)</a>), and if so, whether reads, writes, or both generate the data trace messages.</p> <p>RWT0 values:</p> <p>0b00 = No CDC data trace messages generated for ETPU window 0</p> <p>0b01 = Enable CDC data read trace for ETPU window 0</p> <p>0b10 = Enable CDC data write trace for ETPU window 0</p> <p>0b11 = Enable CDC data read and write trace for ETPU window 0</p> |
| RWT1  | <p>CDC Read/Write Trace 1 Control</p> <p>RWT1 controls whether data trace messages are generated for CDC accesses inside ETPU data trace window 1 (see <a href="#">Data Trace Address Range 1 Register (NDEDI_DTAR1)</a>), and if so, whether reads, writes, or both generate the data trace messages.</p> <p>RWT1 values:</p> <p>0b00 = No CDC data trace messages generated for ETPU window 1</p> <p>0b01 = Enable CDC data read trace for ETPU window 1</p> <p>0b10 = Enable CDC data write trace for ETPU window 1</p> <p>0b11 = Enable CDC data read and write trace for ETPU window 1</p> |
| RWT2  | <p>CDC Read/Write Trace 2 Control</p> <p>RWT2 controls whether data trace messages are generated for CDC accesses inside ETPU data trace window 2 (see <a href="#">Data Trace Address Range 2 Register (NDEDI_DTAR2)</a>), and if so, whether reads, writes, or both generate the data trace messages.</p> <p>RWT2 values:</p> <p>0b00 = No CDC data trace messages generated for ETPU window 2</p> <p>0b01 = Enable CDC data read trace for ETPU window 2</p> <p>0b10 = Enable CDC data write trace for ETPU window 2</p> <p>0b11 = Enable CDC data read and write trace for ETPU window 2</p> |
| RWT3  | <p>CDC Read/Write Trace 3 Control</p> <p>RWT3 controls whether data trace messages are generated for CDC accesses inside ETPU data trace window 3 (see <a href="#">Data Trace Address Range 3 Register (NDEDI_DTAR3)</a>), and if so, whether reads, writes, or both generate the data trace messages.</p> <p>RWT3 value:</p> <p>0b00 = No CDC data trace messages generated for ETPU window 3</p> <p>0b01 = Enable CDC data read trace for ETPU window 3</p> <p>0b10 = Enable CDC data write trace for ETPU window 3</p> <p>0b11 = Enable CDC data read and write trace for ETPU window 3</p>  |
| RC0   | <p>Range Control 0</p> <p>Controls which addresses match data trace window 0 .</p> <p>0 = Trace address inside (inclusive) of data trace window 0</p> <p>1 = Trace addresses outside (exclusive) of data trace window 0</p>                                                                                                                                                                                                                                                                                                                                                                     |

*Table continues on the next page...*

**Table 53-19. NDEDI\_CDC\_DTC field description (continued)**

| Field | Description                                                                                                                                                                                              |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RC1   | Range Control 1<br>Controls which addresses match data trace window 1 .<br>0 = Trace address inside (inclusive) of data trace window 1<br>1 = Trace addresses outside (exclusive) of data trace window 1 |
| RC2   | Range Control 2<br>Controls which addresses match data trace window 2 .<br>0 = Trace address inside (inclusive) of data trace window 2<br>1 = Trace addresses outside (exclusive) of data trace window 2 |
| RC3   | Range Control 3<br>Controls which addresses match data trace window 3 .<br>0 = Trace address inside (inclusive) of data trace window 3<br>1 = Trace addresses outside (exclusive) of data trace window 3 |

### 53.6.1.16 Data Trace Address Range 0 Register (NDEDI\_DTAR0)

The NDEDI\_DTAR0 register defines a data trace address range. The address range is shared by the two ENGINES and the CDC. All three of these clients can independently enable the range for reads and/or writes. If the start address value is greater than the end address value, all accesses are considered to have the address value outside to the data trace window 0.<sup>2</sup>

Notice that the combination of the start and end addresses for data trace into one register does not follow the standard's recommended method of having separate registers for start and end addresses.

Client Select: ENGINE1\_SRC or ENGINE2\_SRC or CDC\_SRC

Register index: 65

2. If RC0 equals 1 for a determined source all accesses from that source will have the address value acknowledged.

Register definition

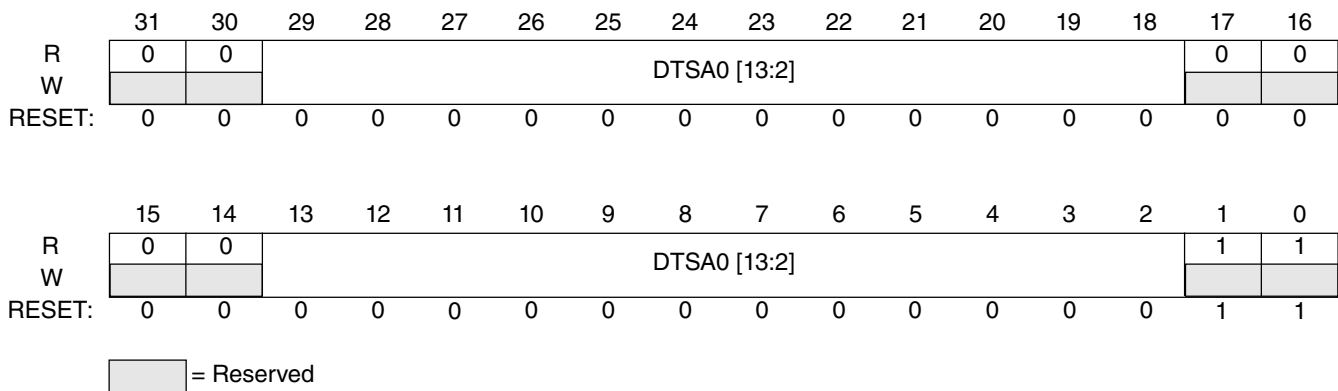


Figure 53-19. Data Trace Address Range 0 Register (NDEDI\_DTAR0)

Table 53-20. NDEDI\_DTAR0 field descriptions

| Field | Description                                                                                      |
|-------|--------------------------------------------------------------------------------------------------|
| DTSA0 | Data Trace Start Address 0<br>The DTSA0 field is the start address for ETPU data trace window 0. |
| DTEA0 | Data Trace End Address 0<br>The DTEA0 field is the end address for ETPU data trace window 0.     |

53.6.1.17 Data Trace Address Range 1 Register (NDEDI\_DTAR1)

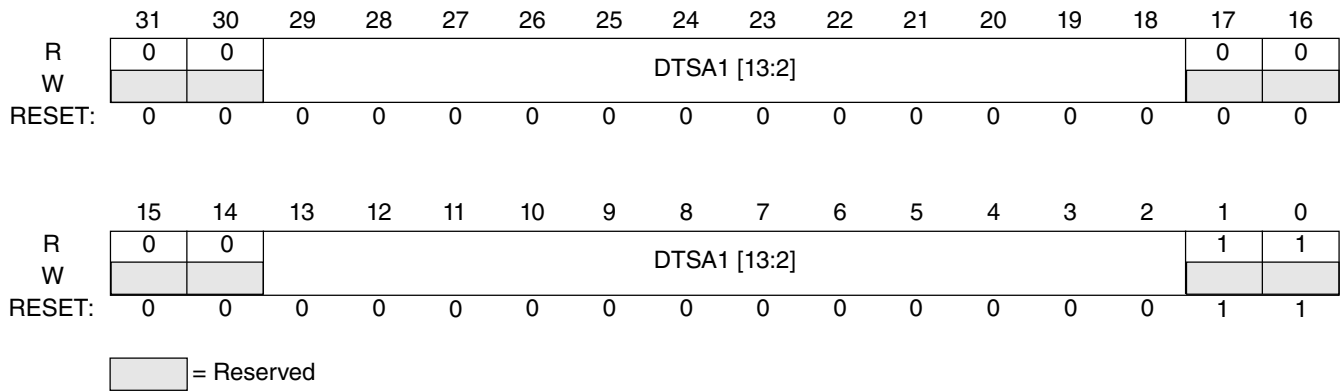
The NDEDI\_DTAR1 register defines a data trace address range. The address range is shared by the two ENGINES and the CDC. All three of these clients can independently enable the range for reads and/or writes. If the start address value is greater than the end address value, all accesses are considered to have the address value outside to the data trace window 1.<sup>3</sup>

Notice that the combination of the start and end addresses for data trace into one register does not follow the standard's recommended method of having separate registers for start and end addresses.

Client Select: ENGINE1\_SRC or ENGINE2\_SRC or CDC\_SRC

Register index: 66

3. If RC1 equals 1 for a determined source all accesses from that source will have the address value acknowledged.

**Figure 53-20. Data Trace Address Range 1 Register (NDEDI\_DTAR1)****Table 53-21. NDEDI\_DTAR1 field descriptions**

| Field | Description                                                                                      |
|-------|--------------------------------------------------------------------------------------------------|
| DTSA1 | Data Trace Start Address 1<br>The DTSA1 field is the start address for ETPU data trace window 1. |
| DTEA1 | Data Trace End Address 1<br>The DTEA1 field is the end address for ETPU data trace window 1.     |

### 53.6.1.18 Data Trace Address Range 2 Register (NDEDI\_DTAR2)

The NDEDI\_DTAR2 register defines a data trace address range. The address range is shared by the two ENGINES and the CDC. All three of these clients can independently enable the range for reads and/or writes. If the start address value is greater than the end address value, all accesses are considered to have the address value outside to the data trace window 2.<sup>4</sup>

Notice that the combination of the start and end addresses for data trace into one register does not follow the standard's recommended method of having separate registers for start and end addresses.

Client Select: ENGINE1\_SRC or ENGINE2\_SRC or CDC\_SRC

Register index: 67

4. If RC2 equals 1 for a determined source all accesses from that source will have the address value acknowledged.

Register definition

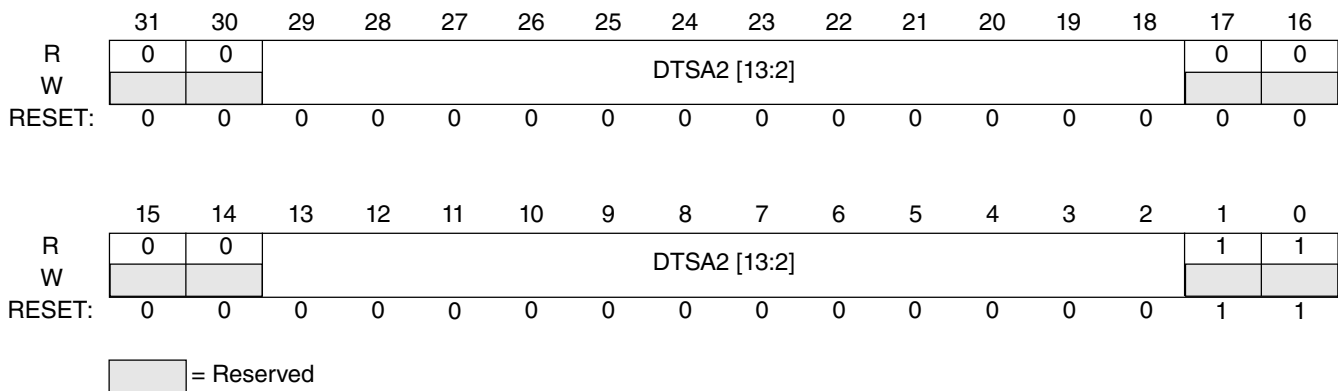


Figure 53-21. Data Trace Address Range 2 Register (NDEDI\_DTAR2)

Table 53-22. NDEDI\_DTAR2 field descriptions

| Field | Description                                                                                      |
|-------|--------------------------------------------------------------------------------------------------|
| DTSA2 | Data Trace Start Address 2<br>The DTSA2 field is the start address for ETPU data trace window 2. |
| DTEA2 | Data Trace End Address 2<br>The DTEA2 field is the end address for ETPU data trace window 2.     |

53.6.1.19 Data Trace Address Range 3 Register (NDEDI\_DTAR3)

The NDEDI\_DTAR3 register defines a data trace address range. The address range is shared by the two ENGINES and the CDC. All three of these clients can independently enable the range for reads and/or writes. If the start address value is greater than the end address value, all accesses are considered to have the address value outside to the data trace window 3.<sup>5</sup>

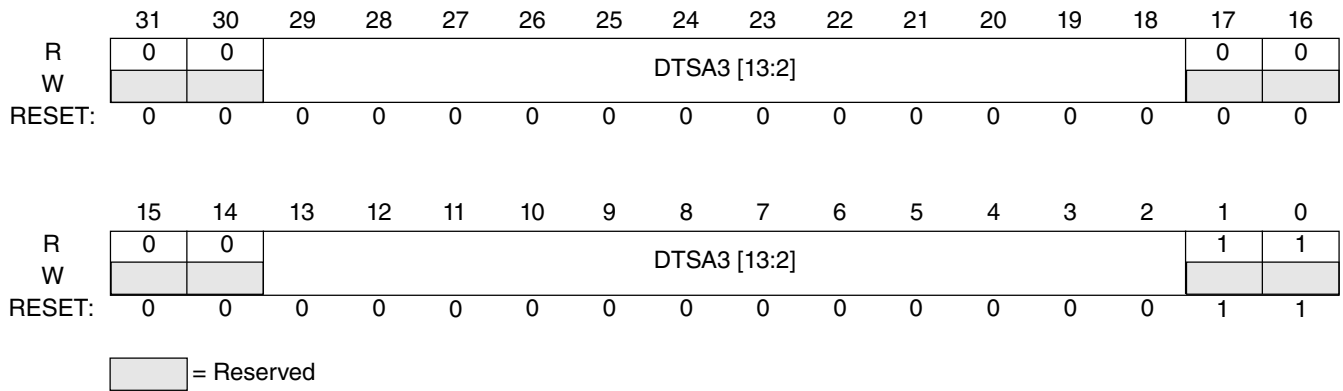
Notice that the combination of the start and end addresses for data trace into one register does not follow the standard's recommended method of having separate registers for start and end addresses.

Client Select: ENGINE1\_SRC or ENGINE2\_SRC or CDC\_SRC

Register index: 68

5. If RC3 equals 1 for a determined source all accesses from that source will have the address value acknowledged.



**Figure 53-22. Data Trace Address Range 3 Register (NDEDI\_DTAR3)****Table 53-23. NDEDI\_DTAR3 field descriptions**

| Field | Description                                                                                           |
|-------|-------------------------------------------------------------------------------------------------------|
| DTSA3 | ETPU Data Trace Start Address 3<br>The DTSA3 field is the start address for ETPU data trace window 3. |
| DTEA3 | ETPU Data Trace End Address 3<br>The DTEA3 field is the end address for ETPU data trace window 3.     |

## 53.6.2 Unimplemented Registers

Unimplemented registers are those with client select and index value combinations other than those listed in [Table 53-3](#). NDEDI will treat unimplemented registers like the JTAG BYPASS instruction.

## 53.7 Functional description

This section describes the functional description.

### 53.7.1 NDEDI reset configuration

This section describes the reset configuration of NDEDI.

### 53.7.1.1 Enabling NDEDI Class 1 operation

The NDEDI Class 1 features are always enabled after exiting the JTAG Test-Logic-Reset state. But they are disabled while the JTAG is in this state. Thus, allowing low power mode for a production part. The registers are always reset while in the JTAG Test-Logic-Reset state.

### 53.7.1.2 Enabling NDEDI Class 3 operation

When NDEDI input `nex_mcko_en` is asserted the NDEDI Classes 1 and 3 are enabled after exiting the JTAG Test-Logic-Reset state, otherwise the NDEDI Class 3 features will be disabled, entering in the Disable-Port Mode, thus no trace output will be provided, and auxiliary port output pins will be disabled (driven inactive or used for an alternate function if sharing pins).<sup>6</sup>

## 53.7.2 Auxiliary output port

This section describes the Auxiliary output ports.

### 53.7.2.1 Output Message Protocol

The protocol for transmitting messages via the auxiliary port is accomplished with the  $\overline{\text{MSEO}}$  functions.  $\overline{\text{MDO}}$  and  $\overline{\text{MSEO}}$  must be sampled by the development tool on the rising edge of  $\text{MCKO}$ .

The following figure illustrates the state diagram for  $\overline{\text{MSEO}}$  transfers. All transitions not included in the figure are reserved, and are never generated by the NDEDI block.

---

6. Class 1 features are still available in Disable-Port Mode.

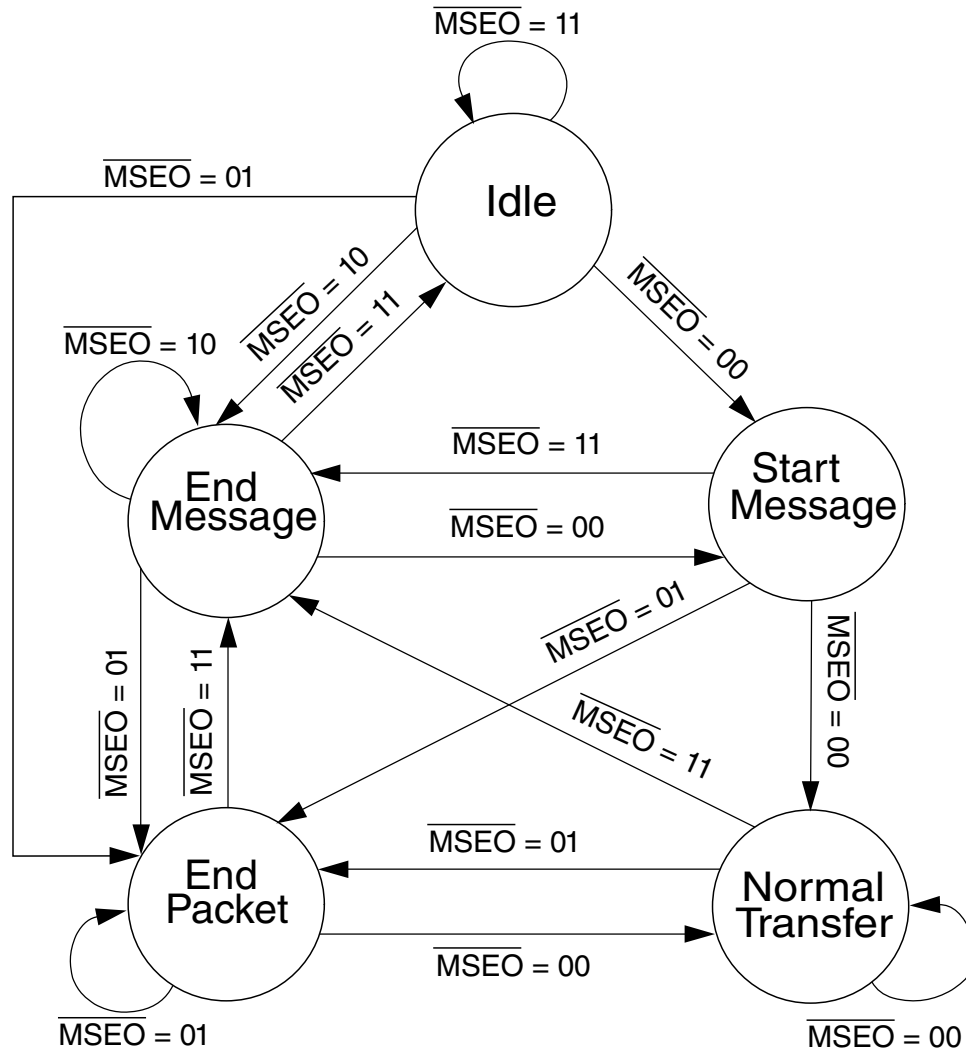


Figure 53-23. MSEO Transfers

### 53.7.2.2 Output messages

The following table describes the messages that the NDEDI can transmit on the auxiliary port.

Table 53-24. NDEDI messages

| Message Name            | Min. Packet Size (bits) | Max Packet Size (bits) | Packet Type | Packet Name | Packet Description                       |
|-------------------------|-------------------------|------------------------|-------------|-------------|------------------------------------------|
| Debug Status Message    | 6                       | 6                      | fixed       | TCODE       | Value = 0                                |
|                         | K                       | K                      | fixed       | SRC         | Client that is the source of the message |
|                         | 32                      | 32                     | fixed       | STATUS      | Value of the Development Status register |
| Ownership Trace Message | 6                       | 6                      | fixed       | TCODE       | Value = 2                                |
|                         | K                       | K                      | fixed       | SRC         | Client that is the source of the message |

Table continues on the next page...

**Table 53-24. NDEDI messages (continued)**

| Message Name                             | Min. Packet Size (bits) | Max Packet Size (bits) | Packet Type | Packet Name | Packet Description                                                                                     |
|------------------------------------------|-------------------------|------------------------|-------------|-------------|--------------------------------------------------------------------------------------------------------|
|                                          | 16                      | 16                     | fixed       | PROCESS     | Process ID. This value depends on the ETPU state at the beginning of a service                         |
| Data Trace, Data Write Message           | 6                       | 6                      | fixed       | TCODE       | Value = 5                                                                                              |
|                                          | K                       | K                      | fixed       | SRC         | Client that is the source of the message                                                               |
|                                          | 2                       | 2                      | fixed       | SIZE        | Size of data access (8, 24, or 32 bits)                                                                |
|                                          | 1                       | 14                     | variable    | U-ADDR      | Unique portion of the data write address. Most significant bits that have 0 values may be truncated    |
|                                          | 1                       | 32                     | variable    | DATA        | Data value written                                                                                     |
| Data Trace, Data Read Message            | 6                       | 6                      | fixed       | TCODE       | Value = 6                                                                                              |
|                                          | K                       | K                      | fixed       | SRC         | Client that is the source of the message                                                               |
|                                          | 2                       | 2                      | fixed       | SIZE        | Size of data access (8, 24, or 32 bits)                                                                |
|                                          | 1                       | 14                     | variable    | U-ADDR      | Unique portion of the data read address. Most significant bits that have 0 values may be truncated     |
|                                          | 1                       | 32                     | variable    | DATA        | Data value read                                                                                        |
| Error Message                            | 6                       | 6                      | fixed       | TCODE       | Value = 8                                                                                              |
|                                          | K                       | K                      | fixed       | SRC         | Client that is the source of the message                                                               |
|                                          | 5                       | 5                      | fixed       | ECODE       | Error Code                                                                                             |
| Data Trace, Data Write with Sync Message | 6                       | 6                      | fixed       | TCODE       | Value = 13                                                                                             |
|                                          | K                       | K                      | fixed       | SRC         | Client that is the source of the message                                                               |
|                                          | 2                       | 2                      | fixed       | SIZE        | Size of data access (8, 24, or 32 bits)                                                                |
|                                          | 1                       | 14                     | variable    | F-ADDR      | Full address of the memory location written. Most significant bits that have 0 values may be truncated |
|                                          | 1                       | 32                     | variable    | DATA        | Data value written                                                                                     |
| Data Trace, Data Read with Sync Message  | 6                       | 6                      | fixed       | TCODE       | Value = 14                                                                                             |
|                                          | K                       | K                      | fixed       | SRC         | Client that is the source of the message                                                               |
|                                          | 2                       | 2                      | fixed       | SIZE        | Size of data access (8, 24, or 32 bits)                                                                |
|                                          | 1                       | 14                     | variable    | F-ADDR      | Full address of the memory location read. Most significant bits that have 0 values may be truncated    |
|                                          | 1                       | 32                     | variable    | DATA        | Data value read                                                                                        |
| Watchpoint Hit Message                   | 6                       | 6                      | fixed       | TCODE       | Value = 15                                                                                             |
|                                          | K                       | K                      | fixed       | SRC         | Client that is the source of the message                                                               |
|                                          | 7                       | 7                      | fixed       | WPHIT       | Number indicating watchpoint source                                                                    |
| Resource Full Message                    | 6                       | 6                      | fixed       | TCODE       | Value = 27                                                                                             |
|                                          | K                       | K                      | fixed       | SRC         | Client that is the source of the message                                                               |
|                                          | 4                       | 4                      | fixed       | RCODE       | Resource Code. Refer to <a href="#">Table 53-26</a>                                                    |

Table continues on the next page...

**Table 53-24. NDEDI messages (continued)**

| Message Name                                             | Min. Packet Size (bits) | Max Packet Size (bits) | Packet Type | Packet Name | Packet Description                                                                                                                                                                                                        |
|----------------------------------------------------------|-------------------------|------------------------|-------------|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                                          | 1                       | H                      | variable    | HIST        | Branch/Predicate Instruction History. This packet is terminated by a stop bit set to 1 after the last history bit. This allows the tool to determine which bits are part of the history packet and which are padded zeros |
| Program Trace, Indirect Branch with History Message      | 6                       | 6                      | fixed       | TCODE       | Value = 28                                                                                                                                                                                                                |
|                                                          | K                       | K                      | fixed       | SRC         | Client that is the source of the message                                                                                                                                                                                  |
|                                                          | 1                       | 8                      | variable    | I-CNT       | Number of instruction units executed since the last taken branch, not taken direct branch, or predicated instruction                                                                                                      |
|                                                          | 1                       | 14                     | variable    | U-ADDR      | Unique portion of the branch target address for an indirect branch. Most significant bits that have 0 values may be truncated                                                                                             |
|                                                          | 1                       | H                      | variable    | HIST        | Branch/Predicate Instruction History. This packet is terminated by a stop bit set to 1 after the last history bit. This allows the tool to determine which bits are part of the history packet and which are padded zeros |
| Program Trace, Indirect Branch with History Sync Message | 6                       | 6                      | fixed       | TCODE       | Value = 29                                                                                                                                                                                                                |
|                                                          | K                       | K                      | fixed       | SRC         | Client that is the source of the message                                                                                                                                                                                  |
|                                                          | 1                       | 8                      | variable    | I-CNT       | Number of instruction units executed since the last taken branch, not taken direct branch, or predicated instruction                                                                                                      |
|                                                          | 1                       | 14                     | variable    | F-ADDR      | Full branch target address for an indirect branch. Most significant bits that have 0 values may be truncated                                                                                                              |
|                                                          | 1                       | H                      | variable    | HIST        | Branch/Predicate Instruction History. This packet is terminated by a stop bit set to 1 after the last history bit. This allows the tool to determine which bits are part of the history packet and which are padded zeros |
| Program Trace Correlation Message                        | 6                       | 6                      | fixed       | TCODE       | Value = 33                                                                                                                                                                                                                |
|                                                          | K                       | K                      | fixed       | SRC         | Client that is the source of the message                                                                                                                                                                                  |
|                                                          | 4                       | 4                      | fixed       | EV-CODE     | Event Code                                                                                                                                                                                                                |
|                                                          | 1                       | 8                      | variable    | I-CNT       | Number of instruction units executed since the last taken branch, not taken direct branch, or predicated instruction                                                                                                      |
|                                                          | 1                       | H                      | variable    | HIST        | Branch/Predicate Instruction History. This packet is terminated by a stop bit set to 1 after the last history bit. This allows the tool to determine which bits are part of the history packet and which are padded zeros |
| Program Trace, Channel Start Service Message             | 6                       | 6                      | fixed       | TCODE       | Value = 58 (Vendor-Defined Message)                                                                                                                                                                                       |
|                                                          | K                       | K                      | fixed       | SRC         | Client that is the source of the message                                                                                                                                                                                  |
|                                                          | 5                       | 5                      | fixed       | S-CHAN      | Number of the channel being serviced                                                                                                                                                                                      |

Table continues on the next page...

**Table 53-24. NDEDI messages (continued)**

| Message Name                                | Min. Packet Size (bits) | Max Packet Size (bits) | Packet Type | Packet Name | Packet Description                                                                                                                                                                                                                                        |
|---------------------------------------------|-------------------------|------------------------|-------------|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                             | 1                       | 8                      | variable    | I-CNT       | Number of instruction units executed since the last taken branch, not taken direct branch, or predicated instruction                                                                                                                                      |
|                                             | 1                       | 14                     | variable    | F-ADDR      | Full address of the first instruction. Most significant bits that have 0 values may be truncated.                                                                                                                                                         |
|                                             | 1                       | H                      | variable    | HIST        | Branch/Predicate Instruction History. This packet is terminated by a stop bit set to 1 after the last history bit. This allows the tool to determine which bits are part of the history packet and which are padded zeros                                 |
| Program Trace, Channel Trace Enable Message | 6                       | 6                      | fixed       | TCODE       | Value = 59 (Vendor-Defined Message)                                                                                                                                                                                                                       |
|                                             | K                       | K                      | fixed       | SRC         | Client that is the source of the message                                                                                                                                                                                                                  |
|                                             | 5                       | 5                      | fixed       | S-CHAN      | Number of the channel being serviced                                                                                                                                                                                                                      |
|                                             | 1                       | 14                     | variable    | F-ADDR      | Full address of the first instruction after the program trace is enabled by a watchpoint. Or, the full address of the first instruction after the Event Queue gets empty after a Queue Overrun. Most significant bits that have 0 values may be truncated |
|                                             | 0                       | 5                      | variable    | CHAN        | Channel Register Value. Only sent if different from S-CHAN value                                                                                                                                                                                          |
| Channel Register Write Message              | 6                       | 6                      | fixed       | TCODE       | Value = 60 (Vendor-Defined Message)                                                                                                                                                                                                                       |
|                                             | K                       | K                      | fixed       | SRC         | Client that is the source of the message                                                                                                                                                                                                                  |
|                                             | 1                       | 5                      | variable    | CHAN        | Value written to the channel register                                                                                                                                                                                                                     |

The following table describes the Error Code encodings for the error messages generated by the NDEDI block.

**Table 53-25. Error Codes Encodings(ECODE)**

| Error Code | Description                                                                                          |
|------------|------------------------------------------------------------------------------------------------------|
| 0b00000    | Ownership trace overrun                                                                              |
| 0b00001    | Program trace overrun                                                                                |
| 0b00010    | Data trace overrun                                                                                   |
| 0b00110    | Watchpoint Overrun                                                                                   |
| 0b00111    | Program and/or Data and/or Ownership Trace Overrun                                                   |
| 0b01000    | Program Trace and/or Data Trace and/or Ownership Trace and/or Watchpoint Overrun                     |
| 0b11000    | Debug Status Overrun                                                                                 |
| 0b11001    | Debug Status Overrun and/or Program and/or Data and/or Ownership Trace Overrun                       |
| 0b11010    | Debug Status and/or Program Trace and/or Data Trace and/or Ownership Trace and/or Watchpoint Overrun |

The following table describes the Resource Code encodings for the resource full messages.

**Table 53-26. Resource Codes Encodings**

| Resource Code | Resource                                | DATA Packet Count | DATA Packet Value                                                                                                                                                                                                               |
|---------------|-----------------------------------------|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0b0000        | Program Trace, Sequential Counter       | 1                 | Branch/Predicate Instruction History. This packet is terminated by a stop bit set to 1 after the last history bit. The value that causes the counter to fill is always the same and can be determined without sending the value |
| 0b0001        | Program Trace, Branch/Predicate History | 1                 | Branch/Predicate Instruction History. This packet is terminated by a stop bit set to 1 after the last history bit                                                                                                               |

Message formatting is performed by the message formatter block. Raw messages read from the message queues are independent of the number of MDO pins implemented.

The following table shows the various message formats and the location of variable length packets. Notice that for variable-length packets, the transmitted size of the packet is determined from the range of the least significant bit to the most significant non-zero-valued bit (i.e. most significant 0 value bits are not transmitted).

**Table 53-27. Message Packet Sizes**

| Message                                  | TCODE | Packet #1 | Packet #2              | Packet #3                       | Packet #4                         | Packet #5 | Packet #6 | Min. Size <sup>1</sup> (bits) | Max Size <sup>2</sup> (bits) |
|------------------------------------------|-------|-----------|------------------------|---------------------------------|-----------------------------------|-----------|-----------|-------------------------------|------------------------------|
| Debug Status Message                     | 0     | Fixed = K | Fixed = 32             | NA                              | NA                                | NA        | NA        | 38+K                          | 38+K                         |
| Ownership Trace Message                  | 2     | Fixed = K | Fixed = 16             | NA                              | NA                                | NA        | NA        | 22+K                          | 22+K                         |
| Data Trace, Data Write Message           | 5     | Fixed = K | Fixed <sup>3</sup> = 2 | Variable<br>Min = 1<br>Max = 14 | Variable<br>Min = 1<br>Max = 32   | NA        | NA        | 10+K                          | 54+K                         |
| Data Trace, Data Read Message            | 6     | Fixed = K | Fixed <sup>3</sup> = 2 | Variable<br>Min = 1<br>Max = 14 | Variable<br>Min = 1<br>Max = 32   | NA        | NA        | 10+K                          | 54+K                         |
| Error Message                            | 8     | Fixed = K | Fixed = 5              | NA                              | NA                                | NA        | NA        | 11+K                          | 11+K                         |
| Data Trace, Data Write with Sync Message | 13    | Fixed = K | Fixed <sup>3</sup> = 2 | Variable<br>Min = 1<br>Max = 14 | Variable<br>Min = 1<br>Max = 32 q | NA        | NA        | 10+K                          | 54+K                         |

*Table continues on the next page...*

**Table 53-27. Message Packet Sizes (continued)**

| Message                                                | TCODE | Packet #1 | Packet #2                      | Packet #3                       | Packet #4                       | Packet #5                      | Packet #6 | Min. Size <sup>1</sup> (bits) | Max Size <sup>2</sup> (bits) |
|--------------------------------------------------------|-------|-----------|--------------------------------|---------------------------------|---------------------------------|--------------------------------|-----------|-------------------------------|------------------------------|
| Data Trace, Data Read with Sync Message                | 14    | Fixed = K | Fixed <sup>3</sup> = 2         | Variable<br>Min = 1<br>Max = 14 | Variable<br>Min = 1<br>Max = 32 | NA                             | NA        | 10+K                          | 54+K                         |
| Watchpoint Hit Message                                 | 15    | Fixed = K | Fixed = 7                      | NA                              | NA                              | NA                             | NA        | 13+K                          | 13+K                         |
| Resource Full Message                                  | 27    | Fixed = K | Fixed = 4                      | Variable<br>Min = 1<br>Max = H  | NA                              | NA                             | NA        | 11+K                          | 10+H+K                       |
| Program Trace, Indirect Branch w/ History Message      | 28    | Fixed = K | Variable<br>Min = 1<br>Max = 8 | Variable<br>Min = 1<br>Max = 14 | Variable<br>Min = 1<br>Max = H  | NA                             | NA        | 9+K                           | 28+H+K                       |
| Program Trace, Indirect Branch w/ History Sync Message | 29    | Fixed = K | Variable<br>Min = 1<br>Max = 8 | Variable<br>Min = 1<br>Max = 14 | Variable<br>Min = 1<br>Max = H  | N/A                            | N/A       | 9+K                           | 28+H+K                       |
| Program Trace, Correlation Message                     | 33    | Fixed = K | Fixed=4                        | Variable<br>Min = 1<br>Max = 8  | Variable<br>Min = 1<br>Max = H  | NA                             | NA        | 12+K                          | 18+H+K                       |
| Program Trace, Channel Start Service Message           | 58    | Fixed = K | Fixed = 5                      | Variable<br>Min = 1<br>Max = 8  | Variable<br>Min = 1<br>Max = 14 | Variable<br>Min = 1<br>Max = H | NA        | 14+K                          | 33+H+K                       |
| Program Trace, Channel Trace Enable Message            | 59    | Fixed = K | Fixed=5                        | Variable<br>Min = 1<br>Max = 14 | Variable<br>Min = 0<br>Max = 5  | NA                             | NA        | 12+K                          | 30+K                         |
| Channel Register Write Message                         | 60    | Fixed = K | Variable<br>Min = 1<br>Max = 5 | NA                              | NA                              | NA                             | NA        | 7+K                           | 11+K                         |

1. Minimum information size. The actual number of bits transmitted depends on the number of MDO pins.
2. Maximum information size. The actual number of bits transmitted depends on the number of MDO pins.
3. Packet is fixed and NDEDI always transmits 2 bits, but other sources sharing the auxiliary port may have a different packet size{ndedi\_msg\_checker, ndedi\_dt\_rnd}. Packet size can always be determined by the value of the SRC packet.  
{Statement}



The double edges in the preceding table indicate the starts and ends of messages. The shaded edges in the preceding figure indicate the end of a variable length packet which is not also the end of the message. Packets without shaded areas between them are grouped into super packets and are transmitted together without end-of-packet indications between them.

#### 53.7.2.2.1 Rules of messaging

- A variable-length packet within a message must end on a port boundary (Port boundaries depend on the number of MDO pins active with the current reset configuration)
- A variable-length packet may start within a port boundary only when following a fixed-length packet.
- Superblocks must end on a port boundary
- When a variable-length packet is sized such that it does not end on a port boundary, it is necessary to extend and zero fill the remaining bits after the highest order bit so that it can end on a port boundary
- Multiple fixed-length packets may start and/or end on the same clock
- When any packet follows a variable-length packet, it must start on a port boundary
- The packet containing the TCODE number is always transferred out first, followed by subsequent packets of information
- Within a packet, the least significant bits are shifted out first. The following figure shows the transmission sequence of a message that is made up of a TCODE followed by two packets

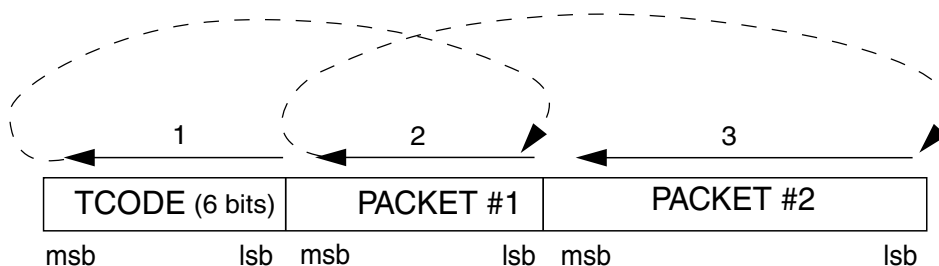


**Figure 53-24. Transmission Sequence of Messages**

#### 53.7.2.2.2 Examples

The following are examples of branch trace and data trace messages.

Functional description

The following figure illustrates how the indirect branch with history public message is transmitted. *Notice that the example uses a 4-bit MDO port* . For this example, 4 bits are shifted out in the I-CNT packet, 5 bits are shifted out in the HIST packet, and 11 bits are shifted out for the UADDR packet. Notice also that T0, S0, H0, I0, and A0 are the least significant bits where:

- Tx= TCODE packet bits (Fixed)
- Sx= SRC packet bits (Fixed)
- Hx = HIST packet bits (Variable)
- Ix= I-CNT packet bits (Variable)
- Ax= UADDR packet bits (Variable)

| Clock | MDO[3:0] |     |    |      | MSEO[1:0] | State               |
|-------|----------|-----|----|------|-----------|---------------------|
|       | 3        | 2   | 1  | 0    |           |                     |
| 0     | X        | X   | X  | X    | X         | End Message or Idle |
| 1     | T3       | T2  | T1 | T0   | 00        | Start Message       |
| 2     | S1       | S0  | T5 | T4   | 00        | Normal Transfer     |
| 3     | I2       | I1  | I0 | S2   | 00        | Normal Transfer     |
| 4     | 0        | 0   | 0  | I3   | 01        | End Packet          |
| 5     | A3       | A2  | A1 | A0   | 00        | Normal Transfer     |
| 6     | A7       | A6  | A5 | A4   | 00        | Normal Transfer     |
| 7     | 0        | A10 | A9 | A8   | 01        | End Packet          |
| 8     | H3       | H2  | H1 | H0   | 00        | Normal Transfer     |
| 9     | 0        | 0   | 0  | H4=1 | 11        | End Message         |

Figure 53-25. Indirect Branch with History Message (4 MDO pins)

The following figure shows the same message being sent using 8 MDO pins

| Clock | MDO[7:0] |    |    |      |    |     |    |    | $\overline{\text{MSEO}}[1:0]$ | State               |
|-------|----------|----|----|------|----|-----|----|----|-------------------------------|---------------------|
|       | 7        | 6  | 5  | 4    | 3  | 2   | 1  | 0  |                               |                     |
| 0     | X        | X  | X  | X    | X  | X   | X  | X  | X                             | End Message or Idle |
| 1     | S1       | S0 | T5 | T4   | T3 | T2  | T1 | T0 | 00                            | Start Message       |
| 2     | 0        | 0  | 0  | I3   | I2 | I1  | I0 | S2 | 01                            | End Packet          |
| 4     | A7       | A6 | A5 | A4   | A3 | A2  | A1 | A0 | 00                            | Normal Transfer     |
| 5     | 0        | 0  | 0  | 0    | 0  | A10 | A9 | A8 | 01                            | End Packet          |
| 3     | 0        | 0  | 0  | H4=1 | H3 | H2  | H1 | H0 | 11                            | End Message         |

Figure 53-26. Indirect Branch with History Message (8 pins)

The following figure illustrates how a data write with sync message is transmitted. *Notice that the example uses a 16-bit MDO port*. For this example, 7 bits are sent for the F-ADDR packet, and 5 bits are sent for the DATA packet. Notice also that T0, S0, A0, and D0 are the least significant bits where:

Tx= TCODE packet bits (Fixed)

Sx= SRC packet bits (Fixed)

Zx = SIZE packet bits (Fixed)

Dx= DATA packet bits (Variable)

Ax= F-ADDR packet bits (Variable)

| Clock | MDO[15:0] |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | $\overline{\text{MSEO}}[1:0]$ | State               |
|-------|-----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------------------------|---------------------|
|       | 15        | 14 | 13 | 12 | 11 | 10 | 9  | 8  | 7  | 6  | 5  | 4  | 3  | 2  | 1  | 0  |                               |                     |
| 0     | X         | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X                             | End Message or Idle |
| 1     | A4        | A3 | A2 | A1 | A0 | Z1 | Z0 | S2 | S1 | S0 | T5 | T4 | T3 | T2 | T1 | T0 | 00                            | Start Message       |
| 2     | 0         | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | A6 | A5 | 01                            | End Packet          |
| 3     | 0         | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | D4 | D3 | D2 | D1 | D0 | 11                            | End Message         |

Figure 53-27. Data Write with Sync Message (16 pins)

### 53.7.2.2.3 Temporal Ordering of Transmitted Messages

All Messages are sent out in the sequence their related event actually occurred. The ETPU interface with the NDEDI provides the ETPU signals to be monitored and captured. At the occurrence of any event that would cause a message, a snapshot of all information needed by the Message Formatter to generate a message is queued. Doing so, several events from either Engines or CDC can be queued at the same time.

Whenever the auxiliary port is idle, the Event Queue is read and the appropriate messages are formatted. Then, NDEDI sends a request for the auxiliary port, and as soon as the auxiliary port is granted the message is sent. If more than one messages are to be sent, the Message Formatter respects the following priority: Messages related to Error, Debug Status, Watchpoints, Program Trace and Data Trace. A request for the auxiliary port is sent for every messages within the snapshot.

NDEDI maintain temporal ordering of messages relative to ENGINE1, ENGINE2 and CDC, but not relative to messages generated by other clients sharing the auxiliary port.

The Time Slot Transition (TST) is considered to be an active state, and since data are accessed at the SPRAM, it is possible that Data Trace Events are recognized within TST.

<sup>7</sup> For further information about TST, please refer to the ETPU BG.

## 53.7.3 IEEE 1149.1 (JTAG) Input Port

The NDEDI block uses the IEEE 1149.1 TAP for accessing registers. This port is shared among all TAP controllers on the MCU. This sharing is controlled by the value in the instruction register and client select control register.

The NDEDI implements a Nexus controller state machine that transitions based on the state of the IEEE 1149.1 state machine shown in [Figure 53-29](#) . The Nexus controller state machine, shown in [Figure 53-30](#), is defined by the IEEE-ISTO 5001-2002 standard.

The instructions implemented by the NDEDI TAP controller are listed in the following table. If any other instruction is received it will be treated as a BYPASS.

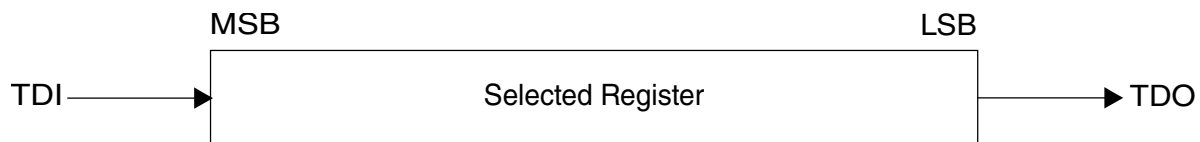
---

7. Due to ETPU operation, speculative accesses may also happen within TST and will be traced.

**Table 53-28. Implemented Instructions**

| Instruction Name | Private/<br>Public | Opcode  | Description                                                                                    |
|------------------|--------------------|---------|------------------------------------------------------------------------------------------------|
| NEXUS-ENABLE     | public             | 4'b0000 | Activate Nexus controller state machine to read and write NDEDI registers                      |
| BYPASS           | public             | 4'b1111 | Implements a single shift register stage providing a minimum serial length between TDI and TDO |

Data is shifted between TDI and TDO starting with the least significant bit, as illustrated in the following figure. This applies for the instruction register and all Nexus tool-mapped registers.

**Figure 53-28. Shifting Data Into Register**

### 53.7.3.1 Enabling the NDEDI TAP Controller

Assertion of the  $\overline{\text{TRST}}$  input resets all TAP controllers on the MCU. The controllers are loaded with the BYPASS instruction upon exit of the 'TEST-LOGIC-RESET' JTAG controller state. Loading the NEXUS-ENABLE instruction grants access to Nexus debug. NDEDI acts as BYPASS by loading any other instruction. The TDO bit is driven with the value of the instruction register while in 'SHIFT\_IR' state.

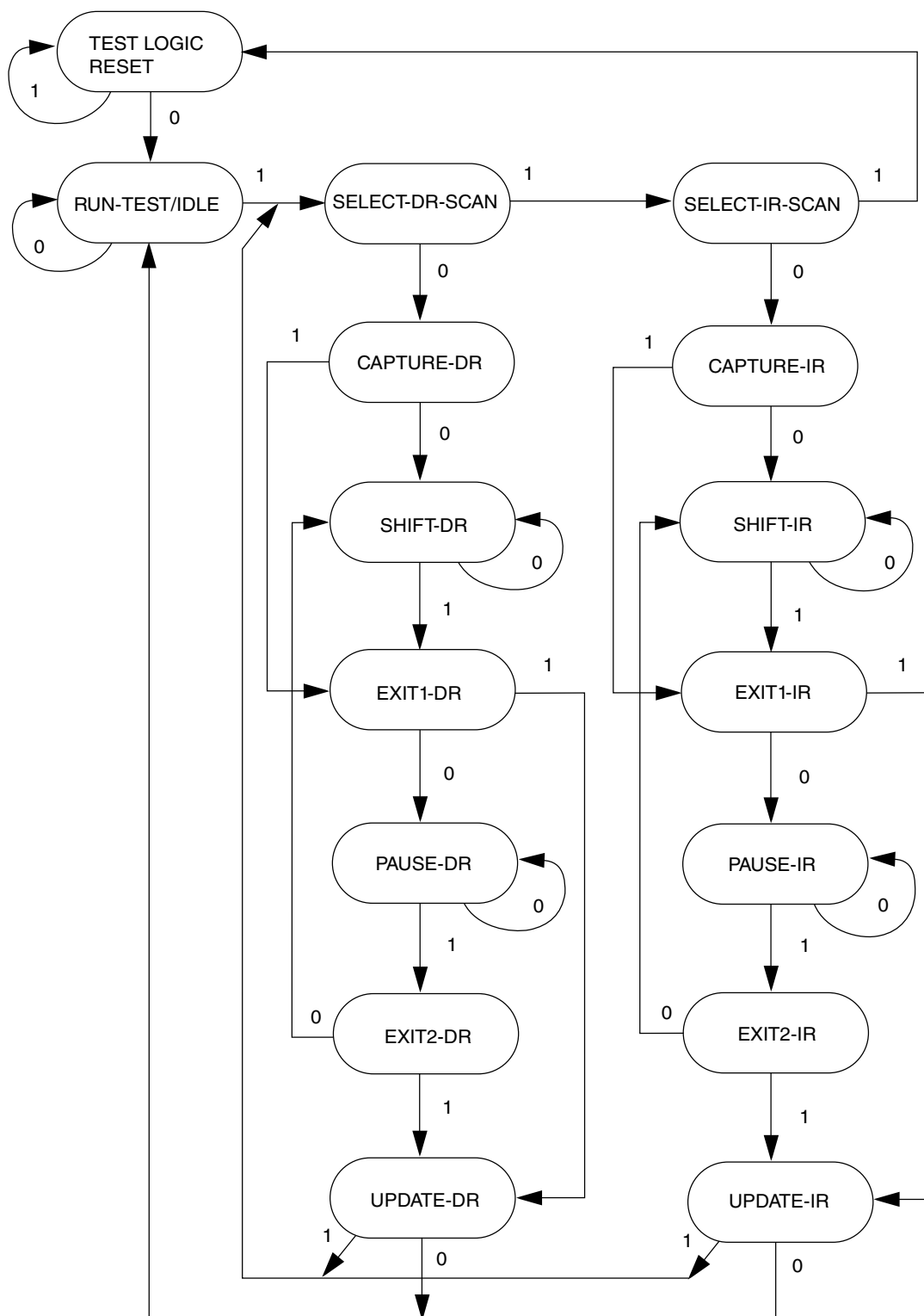


Figure 53-29. IEEE 1149.1 16-State Finite State Machine

### 53.7.3.2 Loading NEXUS-ENABLE Instruction

Access to the NDEDI registers is enabled when the TAP controller instruction register is loaded with the NEXUS-ENABLE instruction. The current instruction value is loaded into the IEEE 1149.1 shifter in the 'CAPTURE-IR' state. The instruction is shifted in via the 'SELECT-IR-SCAN' path and loaded in the 'UPDATE-IR' state. At this point, the Nexus controller state machine, shown in the following figure, transitions to the 'REG\_SELECT' state. The following table illustrates the IEEE 1149.1 sequence to load the NEXUS-ENABLE instruction.

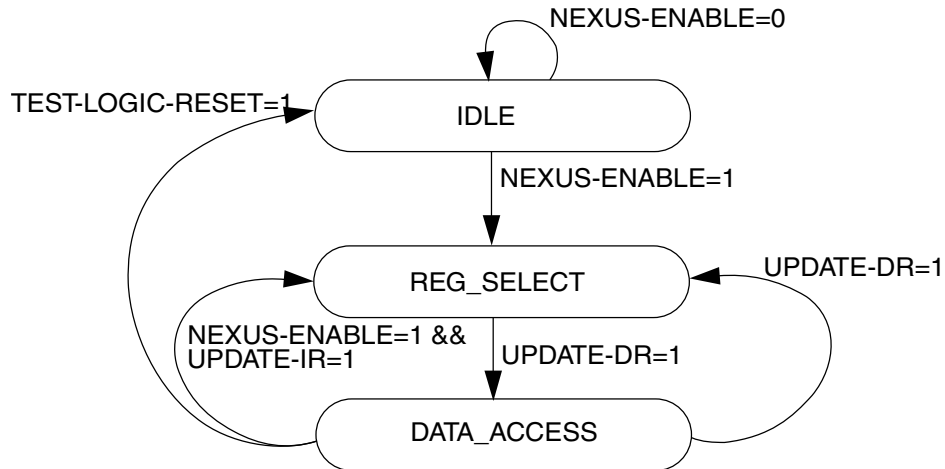


Figure 53-30. Nexus Controller State Machine

Table 53-29. Loading NEXUS-ENABLE instruction

| Clo<br>ck | TM<br>S | IEEE 1149.1 State | Nexus<br>State | Description                                                                                                                |
|-----------|---------|-------------------|----------------|----------------------------------------------------------------------------------------------------------------------------|
| 0         | 0       | RUN-TEST/IDLE     | IDLE           | IEEE 1149.1 controller in idle state                                                                                       |
| 1         | 1       | SELECT-DR-SCAN    | IDLE           | Transitional state                                                                                                         |
| 2         | 1       | SELECT-IR-SCAN    | IDLE           | Transitional state                                                                                                         |
| 3         | 0       | CAPTURE-IR        | IDLE           | Internal shifter loaded with current instruction                                                                           |
| 4         | 0       | SHIFT-IR          | IDLE           | TDO becomes active, and the IEEE 1149.1 shifter is ready.<br>Shift in all but the last bit of the NEXUS_ENABLE instruction |
| 4 TCKs    |         |                   |                |                                                                                                                            |
| 12        | 1       | EXIT1-IR          | IDLE           | Last bit of instruction shifted in                                                                                         |
| 13        | 1       | UPDATE-IR         | IDLE           | NEXUS-ENABLE loaded into instruction register                                                                              |
| 14        | 0       | RUN-TEST/IDLE     | REG_SEL<br>ECT | NEXUS-ENABLE ready to be read                                                                                              |

### 53.7.3.3 Selecting an IEEE 1149.1 Register

When the NEXUS-ENABLE instruction is loaded in the NDEDI TAP controller instruction register, the input port allows development tool access to all NDEDI registers. Each register has a 7-bit address index. Some clients have registers with the same index as registers in other clients. Therefore, to access the registers of a particular client, the Client Select Control register must be set for that client.

All register access is performed via the 'SELECT-DR-SCAN' path. The Nexus state machine defaults to the register select state when enabled. Accessing a register requires two passes through the 'SELECT-DR-SCAN' path: one pass to select the register and the second pass to read/write the register.

The first pass through the 'SELECT-DR-SCAN' path is used to enter an 8-bit Nexus command consisting of a read/write control bit in the LSB followed by a 7-bit register address index, as illustrated in the following figure. The read/write control bit is set to 1 for writes and 0 for reads. The current value of the Controller Command Input is captured into the IEEE 1149.1 shifter during the 'CAPTURE-DR' state while the Nexus controller state machine is in the 'REG\_SELECT' state.

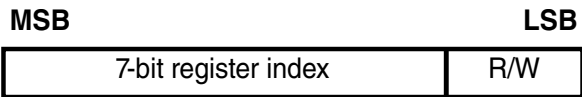


**Figure 53-31. IEEE 1149.1 Controller Command Input**

The second pass through the 'SELECT-DR-SCAN' path is used to read or write the register data by shifting in the data (LSB first) during the 'SHIFT-DR' state. When reading a register, the current register value is loaded into the IEEE 1149.1 shifter during the 'CAPTURE-DR' state. When writing a register, the value is loaded from the IEEE 1149.1 shifter to the register during the 'UPDATE-DR' state. When reading a register, there is no requirement to shift out the entire register contents. Shifting may be terminated once the required number of bits have been acquired.

The following table illustrates a sequence which writes a 32-bit value to a register

**Table 53-30. Writing to a Register**

| Clock | TM S | IEEE 1149.1 State | Nexus State | Description                                                            |
|-------|------|-------------------|-------------|------------------------------------------------------------------------|
| 0     | 0    | RUN-TEST/IDLE     | REG_SELECT  | IEEE 1149.1 controller in idle state                                   |
| 1     | 1    | SELECT-DR-SCAN    | REG_SELECT  | First pass through 'SELECT-DR-SCAN' path                               |
| 2     | 0    | CAPTURE-DR        | REG_SELECT  | Internal shifter loaded with current value of controller command input |

*Table continues on the next page...*



**Table 53-30. Writing to a Register (continued)**

| Clock   | TMS | IEEE 1149.1 State | Nexus State | Description                                                                                           |
|---------|-----|-------------------|-------------|-------------------------------------------------------------------------------------------------------|
| 3       | 0   | SHIFT-DR          | REG_SELECT  | TDO becomes active, and write bit and 6 bits of register index shifted in                             |
| 7 TCKs  |     |                   |             |                                                                                                       |
| 12      | 1   | EXIT1-DR          | REG_SELECT  | Last bit of register index shifted into TDI                                                           |
| 13      | 1   | UPDATE-DR         | REG_SELECT  | Controller decodes and selects register                                                               |
| 14      | 1   | SELECT-DR-SCAN    | DATA_ACCESS | Second pass through 'SELECT-DR-SCAN' path                                                             |
| 15      | 0   | CAPTURE-DR        | DATA_ACCESS | Internal shifter loaded with current value of register                                                |
| 16      | 0   | SHIFT-DR          | DATA_ACCESS | TDO becomes active, and outputs current value of register while new value is shifted in through TDI   |
| 31 TCKs |     |                   |             |                                                                                                       |
| 48      | 1   | EXIT1-DR          | DATA_ACCESS | Last bit of current value shifted out TDO. Last bit of new value shifted in TDI                       |
| 49      | 1   | UPDATE-DR         | DATA_ACCESS | Value written to register                                                                             |
| 50      | 0   | RUN-TEST/IDLE     | REG_SELECT  | Controller returned to idle state. It could also return to 'SELECT-DR-SCAN' to write another register |

### 53.7.4 Microcode Development Support

The NDEDI contains a number of hardware hooks that aid in the development of microcode for the ETPUs. This features described in this section make the ETPUs compliant with Nexus Class 1 features.

The main features provided are the abilities to:

- Read and write ETPU internal registers in debug mode
- Read and write SPRAM in debug mode
- Enter a debug mode at reset negation
- Enter a debug mode during normal execution
- Single step instructions and re-enter debug mode
- Stop program execution on instruction/data breakpoints or channel service breakpoints and enter debug mode
- Set breakpoints or watchpoints

Functional description

- Execute external microcode instruction in debug mode
- Read/write the microprogram counter register value in debug mode

Refer to [Initialization/Application Information](#) on how the following functions provide the features listed above.

53.7.5 Debug Status

The NDEDI module provides the Debug Status via the auxiliary port, as defined by the IEEE-ISTO 5001-2002 standard.

53.7.5.1 Messaging

The NDEDI block provides debug status messaging using IEEE-ISTO 5001-2002 standard-defined public messages. When the development status register changes, a debug status change event is sent to the Event Queue. If the debug status change condition occurs while the Event Queue is not enabled for storing snapshots, a debug status overrun message is generated.

The debug status message has the format shown in the following figure.

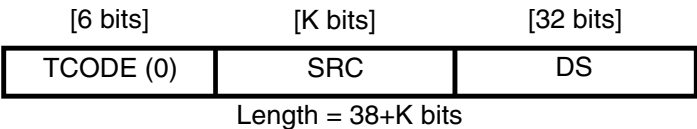


Figure 53-32. Debug Status Message Format

The DS packet correspond to the ENGINE<sub>x</sub> DS register value described in [Figure 53-4](#).

53.7.5.2 Error Messages

A debug status overrun error event is queued if a debug status event occurs while the Event Queue is not enabled for storing snapshots. The error event is stored as soon as the Event Queue is enabled which happens when the queue gets empty.

The error message has the format shown in the following figure.

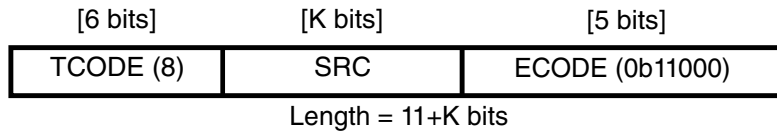


**Figure 53-33. Debug Status Error Message Format**

### 53.7.5.3 Synchronization

Upon the exit of debug mode, the next program and data trace messages are synchronization messages. The exit of debug mode by one of the Engines causes data and program synchronization for that Engine only.

### 53.7.5.4 Timing Diagrams

Note that all of the following timing diagrams assume a 3-bit SRC size and a 16-bit MDO port. The state variable is not a signal, but instead is derived from  $\overline{\text{MSEO}}$ . It is included for clarity. Refer to [Figure 53-23](#) for  $\overline{\text{MSEO}}$  state diagram.

The following abbreviations are used for the state variable in the diagrams:

- ID = Idle
- SM = Start Message
- NT = Normal Transfer
- EP = End Packet
- EM = End Message

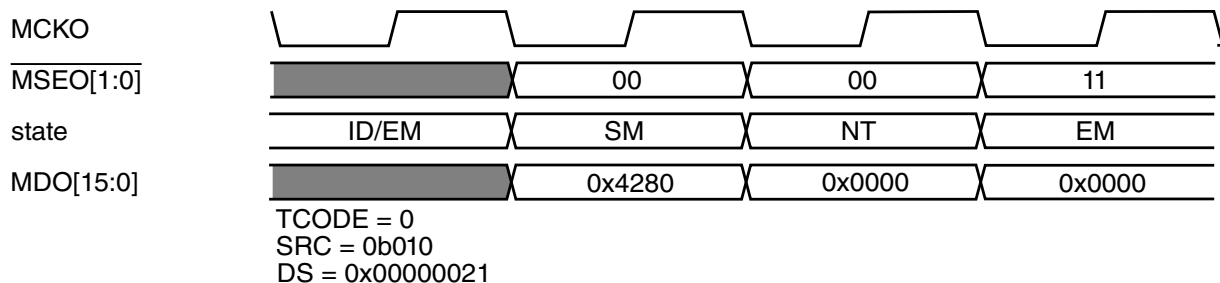


**Figure 53-34. Debug Status Message**

Functional description

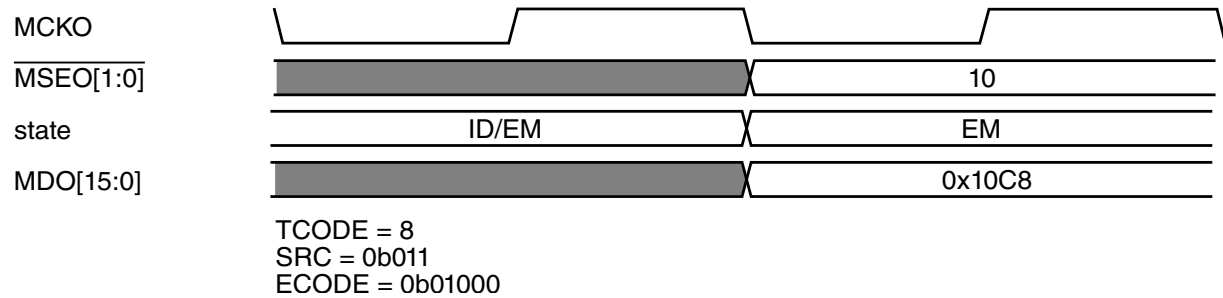


Figure 53-35. Debug Status Overrun Error Message

### 53.7.6 Ownership Trace

Ownership trace provides a macroscopic view of the ETPU program flow. This is done by generating an ownership trace event at the start of each channel service and whenever an engine becomes idle.

Each engine within the ETPU system generates ownership traces independently, and the events are queued in the order in which they occur.

#### 53.7.6.1 Messaging

Ownership trace information is transmitted via the auxiliary port using an ownership trace message (OTM).

Ownership trace information is transmitted in the format shown in the following figure.

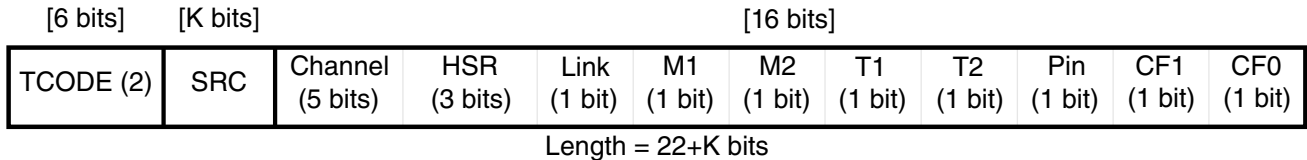


Figure 53-36. Ownership Trace Message Format

Notice that the channel number and entry points number in the preceding figure are concatenated to form the variable length PROCESS packet defined by the IEEE-ISTO 5001-2002 standard.

### 53.7.6.2 OTM Flow

A channel generates an ownership trace message when service starts if ownership trace is enabled for the ETPU via the development control register. If ownership trace is enabled and an engine becomes idle, NDEDI will transmit a special OTM with the PROCESS packet set to 0.

### 53.7.6.3 Timing Diagram

The following figure shows an example of an ownership message being transmitted on the auxiliary port. The diagram assumes a 3-bit SRC size and a 16-bit MDO port. The state variable is not a signal but instead, it is derived from  $\overline{\text{MSEO}}$ . It is included for clarity. Refer to [Figure 53-23](#) for  $\overline{\text{MSEO}}$  state diagram.

The following abbreviations are used for the state variable in the diagrams:

- ID = Idle
- SM = Start Message
- NT = Normal Transfer
- EP = End Packet
- EM = End Message

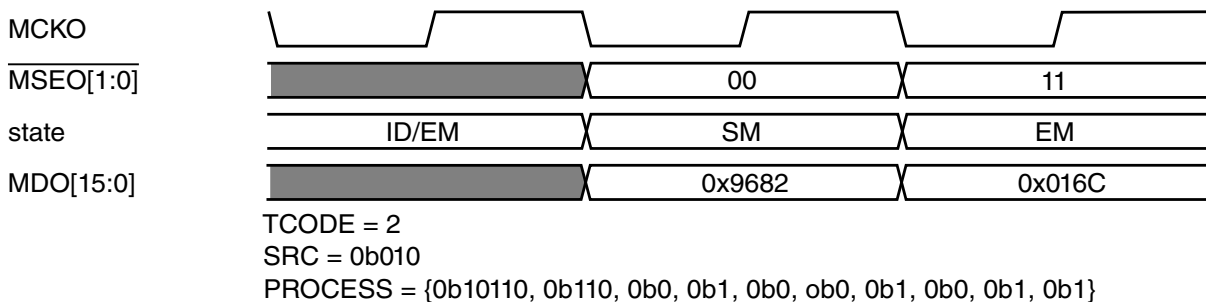


**Figure 53-37. Ownership Trace Message**

### 53.7.7 Program Trace

This section details the program trace mechanism supported by the NDEDI for the dual Engine system. Program trace is implemented via a combination of Branch Trace Messaging (BTM) and Ownership Trace Messaging (OTM) as per the IEEE-ISTO 5001-2002 standard definition.

### 53.7.7.1 Branch Trace Messaging

Branch trace messaging facilitates program trace by providing the following types of information:

- Messages for the start of channel services, each containing the full address of the first instruction to execute and history and sequential count information left over from the previously traced channel service. This message is always synchronizing
- Messages to indicate that the Program Trace was enabled in the middle of a thread or after the Queue is enabled for storing snapshots and it was previously disabled due to queue overrun. It includes the address of the enable point, the value of the serviced channel and the CHAN reg value only if it differs from the serviced channel value
- Messages to indicate that the internal history buffer has filled and been reset. The current value of the history buffer is included in each of these messages
- Messages to indicate that the internal sequential instruction counter has overflow and been reset. The current value of the history buffer is included in each of these messages
- Messages for taken indirect branches, each including a branch/predicate instruction<sup>8</sup> history packet, sequential instruction count, and the unique portion of the branch target address. Indirect branch messages that are generated with pending synchronization event include the full branch target address (instead of the unique portion of the address)
- Messages to indicate the entry into debug mode, the entry into low power mode, or the end of channel service with no new service pending
- Messages for writes to the CHAN register, indicating the value assigned to the register

The address packet of a program trace address excludes the 2 least significant bits of the byte address because they are zero for all instruction addresses.

### 53.7.7.2 Branch Trace Message Formats

There are seven types of messages used by the ETPU for branch tracing. These types are shown in the following table.

---

8. Predicate Instructions depend on the AS/CE field within ETPU instructions.

**Table 53-31. ETPU BTM Messages**

| Message                                              | Synchronizing | Public/Private |
|------------------------------------------------------|---------------|----------------|
| Resource Full Message                                | No            | Public         |
| Indirect Branch with History Message                 | No            | Public         |
| Indirect Branch with History Synchronization Message | Yes           | Public         |
| Program Trace Correlation Message                    | No            | Public         |
| Channel Start Service Message                        | Yes           | Private        |
| Channel Trace Enable Message                         | Yes           | Private        |
| Channel Register Write Message                       | No            | Private        |

The occurrence of any of the following conditions requires subsequent synchronization:

- Exit of system reset
- Exit of low power mode
- Exit of debug mode
- Write to the Development Control Register or the Program Trace Channel Enable Register<sup>9</sup>
- 255 branch trace messages of the same are queued without synchronization
- Assertion of the Event In ( $\overline{\text{EVTI}}$ ) pin, depending on how the development control register is configured
- A watchpoint occurrence<sup>10</sup>
- A queue overrun

The message that follows any of these synchronizing events will be a synchronizing message if it is not a Resource Full Message or a Channel Register Write Message which have no synchronizing information. If the ETPU reaches the idle the next program trace message will be a Channel Start Service Message or a Channel Trace Enable Message, that are synchronizing.

#### 53.7.7.2.1 Resource Full Messages

The resource full message is generated when the branch/predicate history buffer is full or the sequential instruction counter overflows.

9. Only the Source that have the register written will generate a synchronization message.

10. A Watchpoint occurrence for one Engine generates synchronization message for that Engine only.

For messages caused by the branch/predicate history buffer filling, the current value of that buffer is transmitted as part of the resource full message. This information can be concatenated by the development tool with the branch/predicate history information from subsequent messages to obtain the complete branch history for a message. The HIST value is reset by this message, and the I-CNT value is reset as a result of a bit being added to the history buffer.

For messages caused by sequential instruction counter overflow, the HIST packet is also transmitted. This occurrence indicates that exactly 256 sequential instructions have been executed without a history bit being recorded or a message (other than CHAN\_WRITE) being generated. The HIST and I-CNT values are reset by this message.

The resource full message has the format shown in the following figure.

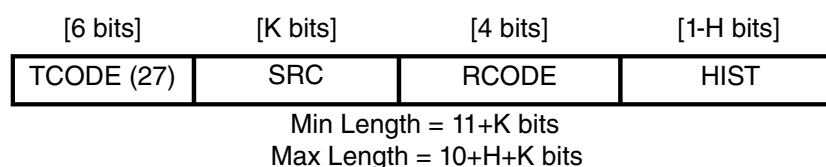


**Figure 53-38. Resource Full Message Format**

### 53.7.7.2.2 Indirect Branch with History Messages

The indirect branch with history message is generated on a taken indirect branch with no synchronization event pending. The program trace indirect branch with history message has the format shown in the following figure.

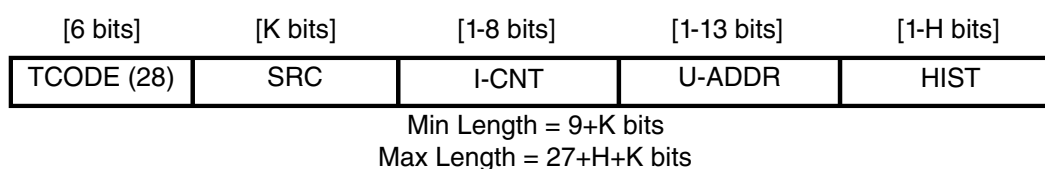


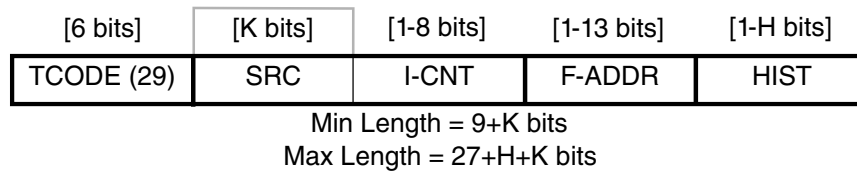
**Figure 53-39. Indirect Branch with History Message Format**

### 53.7.7.2.3 Indirect Branch with History Synchronization Messages

The indirect branch with history synchronization message is generated on a taken indirect branch with a synchronization event pending.

The program trace indirect branch with history synchronization message has the format shown in the following figure.



**Figure 53-40. Indirect Branch with History Synchronization Message Format**

#### 53.7.7.2.4 Program Trace Correlation Message

The program trace correlation message is generated at the end of an ETPU channel service if there is no new service, if there is a new service which channel is not enabled for tracing, if program trace is disabled in the middle of a thread being traced, upon entry into debug mode, or upon entry into low power mode.

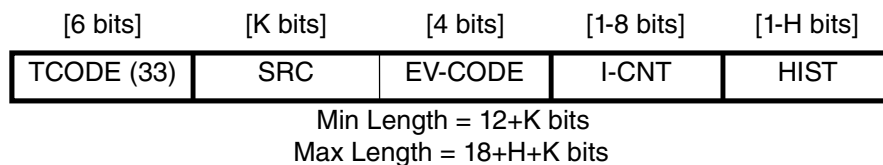
The message contains the history and sequential instruction count information generated during the previous traced service since the last message generation.

The correlation message also contains an event code that indicates the event responsible for the generation of the message. Event code encodings are shown in the following table.

**Table 53-32. Event Code Encodings**

| EV-CODE    | Description                                      |
|------------|--------------------------------------------------|
| 0b0000     | Entry into Debug Mode                            |
| 0b0001     | Entry into Low Power Mode                        |
| 0b0010     | Program Trace disabled in the middle of a thread |
| 0b1110     | End Channel Service (no new service)             |
| all others | Reserved                                         |

The program trace correlation message has the format shown in the following figure.

**Figure 53-41. Program Trace Correlation Message Format**

#### 53.7.7.2.5 Channel Start Service Message

The channel start service message is generated at the start of an ETPU channel service. The message contains the history and sequential instruction count information generated during the previous traced service since the last message generation. The message also contains the address of the first instruction of the current service.

The program trace channel start service message has the format shown in the following figure.

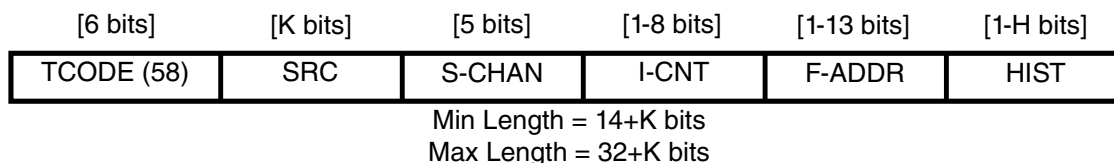


**Figure 53-42. Channel Start Service Synchronization Message Format**

### 53.7.7.2.6 Channel Trace Enable Message

If the program trace is enabled in the middle of a thread or after the Queue is enabled for storing snapshots and it was previously disabled due to queue overrun, the Channel Trace Enable Message is sent indicating to the development tool the number of the channel being serviced and the address of the current instruction.<sup>11</sup> If the CHAN register value differs from the serviced channel, the CHAN field is sent at the end of the message indicating the current CHAN register value.<sup>12</sup>

The program trace channel trace enable message has the format shown in the following figure.

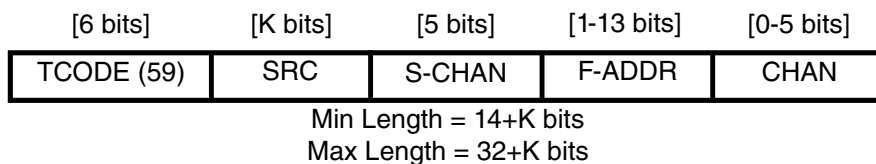


**Figure 53-43. Channel Trace Enable Synchronization Message Format**

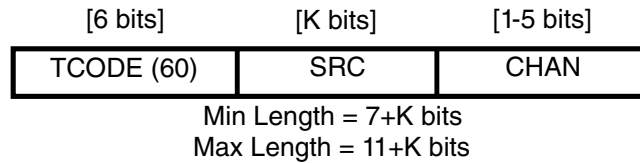
### 53.7.7.2.7 Channel Register Write Messages

The channel register write message is generated when the CHAN register is written to by the microcode. This message is not necessary to reconstruct program flow and generation is enabled/disabled by the CHW bit in the ENGINE<sub>x</sub> development control register. Program trace must be enabled for this message to be generated.

The channel register write message has the format shown in the following figure

11. The message does not contains the history and sequential instruction count information since the trace of this channel was disabled until there.

12. Usually the CHAN register equals the serviced channel.

**Figure 53-44. Channel Register Write Message Format**

### 53.7.7.3 Branch Trace Messaging Operation

Figure 53-45 and Figure 53-46 show the basic flow used to generate branch trace messages for the ETPUs. The following acronyms are used in the figures:

- CHAN - CHAN register in ETPU execution unit
- S-CHAN - Channel being serviced
- HIST - Nexus shift register whose value is transmitted in the HIST packet of many BTMs
- I-CNT - Nexus sequential instruction counter whose value is transmitted in the HIST packet of many BTMs
- CWE - Channel Write Event
- IHE - Indirect Branch with History Event
- IHS - Indirect Branch with History Synchronization Event
- CSE - Channel Start Service Event
- CTE - Channel Trace Enable Event
- RFE - Resource Full Event
- PCE - Program Trace Correlation Event

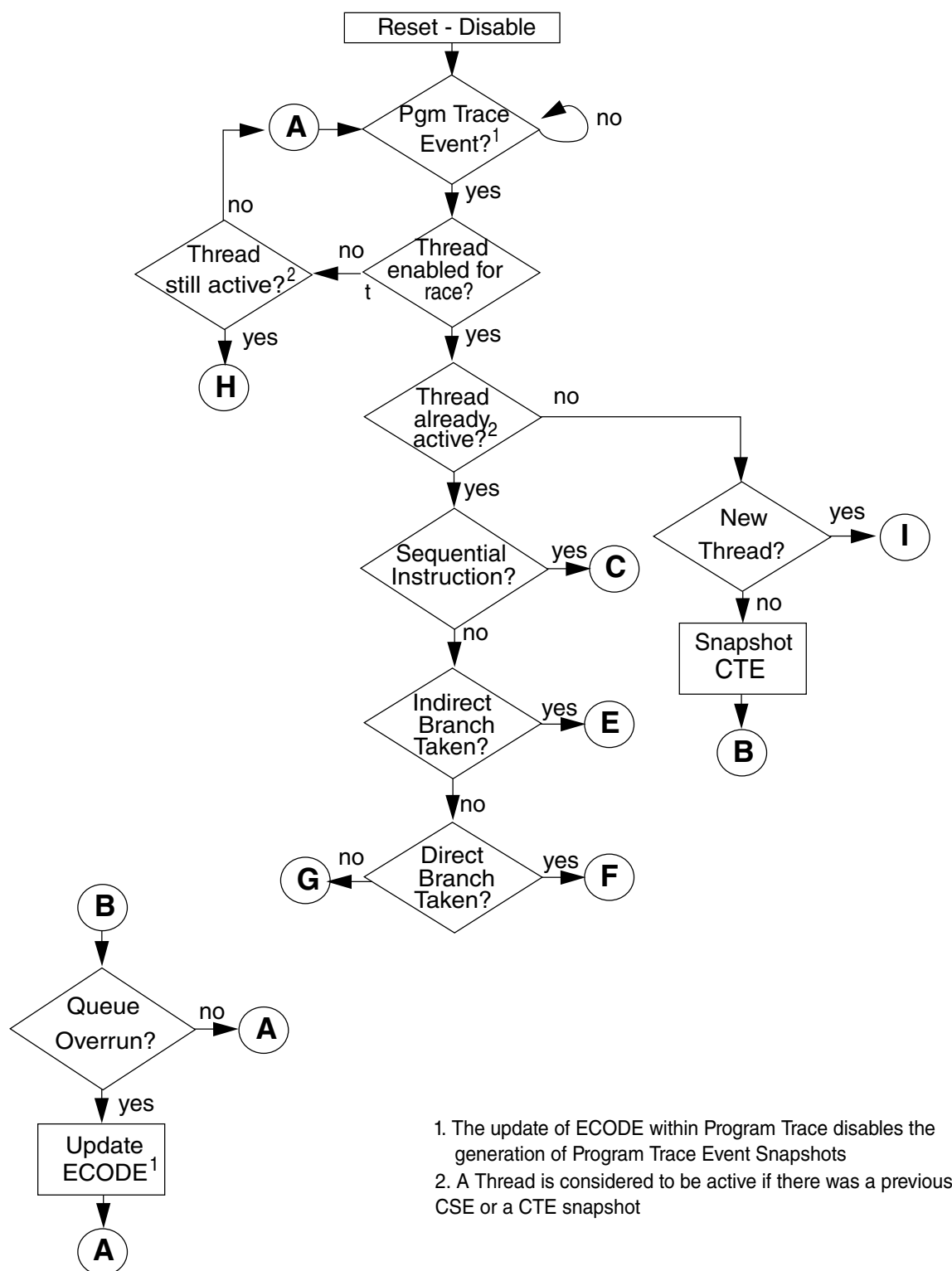


Figure 53-45. Branch Trace Message Generation (Part 1)

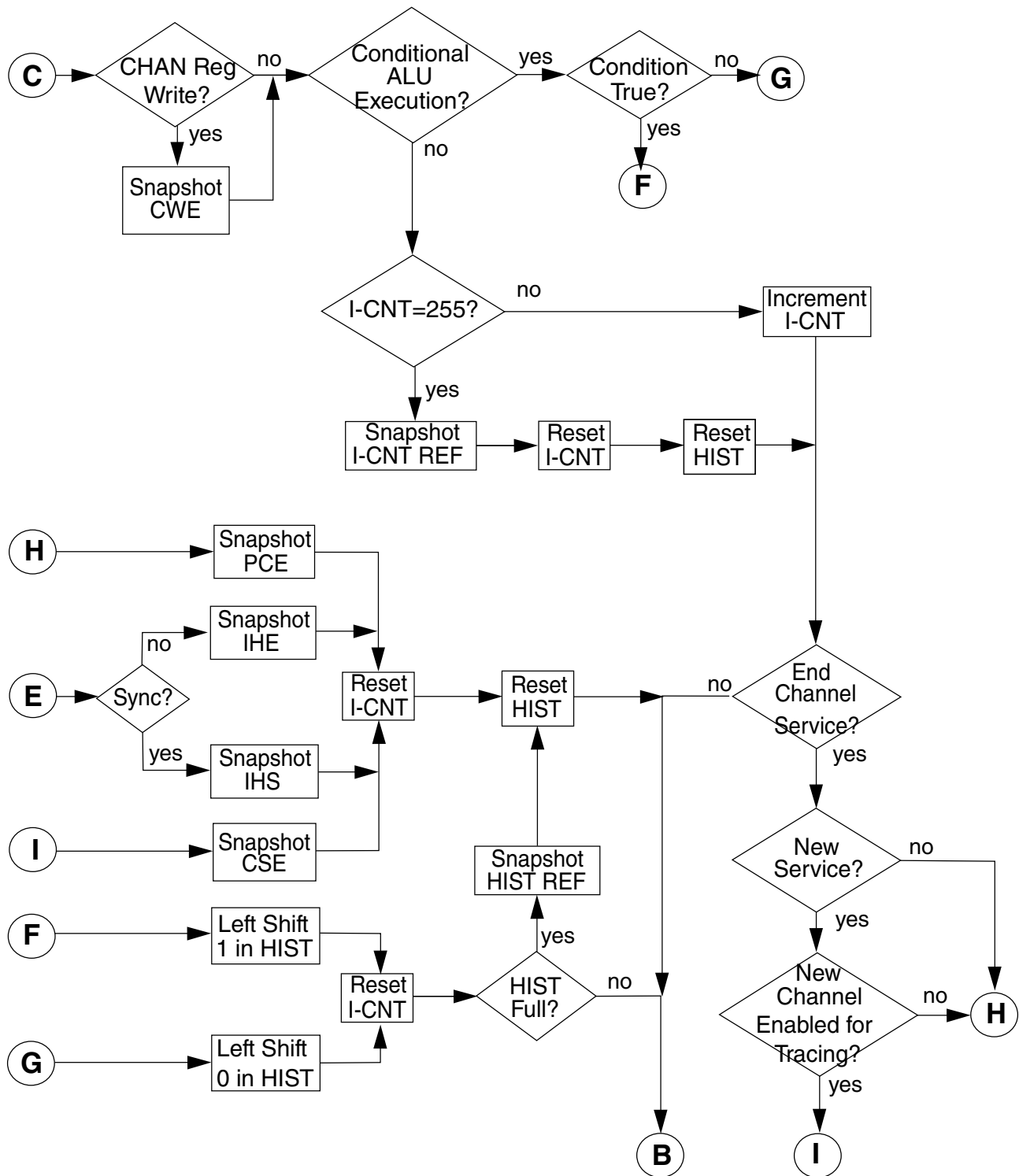


Figure 53-46. Branch Trace Message Generation (Part 2)

### 53.7.7.3.1 Relative Addressing

The relative address feature is compliant with the IEEE-ISTO 5001-2002 standard recommendations and is designed to reduce the number of bits transmitted for addresses of program trace messages.

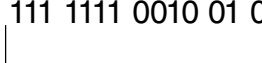
The address transmitted for a determined source is relative to the address of the previous branch trace message sent for that source. It is generated by XORing the new address with the previous address, and then using only the results up to the most significant '1' in the result. To recreate this address, an XOR of the (most-significant 0-padded) message address with the previously decoded address gives the current address. The following figure shows how a relative address is generated and how it can be used to recreate the original address.

Engine 1: Current Address (A1) = 0x65DC,  
 Engine 2: Current Address (B1) = 0x7238

Next Message Relative to Engine 1: Address (A2) = 0x1AF8

### ***Address Generation:***

A1 = 110 0101 1101 11 00  
 A2 = 001 1010 1111 10 00  
 A1 ^ A2 = 111 1111 0010 01 00



Address Message (M1) = 111 1111 0010 01

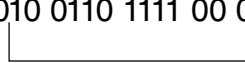
### ***Address Recreation:***

A1 ^ M1 = A2  
 A1 = 110 0101 1101 11 00  
 ME1 = 111 1111 0010 01  
 A1 ^ ME1 = 001 1010 1111 10 00 = A2

Next Message Relative to Engine 2: Address (B2) = 0x54C8

### ***Address Generation for Engine 2.***

B1 = 111 0010 0011 10 00  
 B2 = 101 0100 1100 10 00  
 B1 ^ B2 = 010 0110 1111 00 00



Address Message (ME2) = 10 0110 1111 00

### ***Address Recreation:***

B1 ^ M1 = B2  
 B1 = 111 0010 0011 10 00  
 ME2 = 10 0110 1111 00  
 B1 ^ ME2 = 101 0100 1100 10 00 = B2

**Figure 53-47. Relative Address Generation and Recreation**

#### **53.7.7.3.2 Enabling Program Trace**

Program trace for the ETPU is enabled on a channel-by-channel basis. This selective enabling allows the tool more control over how much information gets transmitted.

For program trace information to be generated for a channel, the appropriate bit in the program trace channel enable register must be set and program trace must be enabled via the TM field in the DC register or via a watchpoint trigger.

Any combination of channels can be enabled for trace at the same time by configuring the program trace channel enable register. See [ENGINEx Program Trace Channel Enable Register \(NDEDI\\_ENGINEx\\_PTCE\)](#) for more details.

Note that since the registers may be changed in the middle of a thread, program trace may be enabled or disabled in the middle of a thread.

If program trace for a determined channel is enabled in the middle of a service routine, a Channel Trace Enable Message is sent to the development tool indicating the number of the channel being serviced and the address of the current instruction. Otherwise, normal program trace begins only on the following service for that channel.

If program trace for a determined channel is disabled in the middle of a service routine, a Program Correlation Message is sent to the development tool indicating that the trace is disabled and informing the HIST and I-CNT values.

### **53.7.7.3.3 Branch/Predicate Instruction History**

The HIST packet provides a history of direct branch and predicated instruction execution used for reconstructing the program flow. This packet is implemented as a left-shifting shift register. The register is always pre-loaded with a value of 1. This bit acts as a stop bit so the tool can determine which bit is the end of the history information. The pre-loaded 1 bit itself is not part of the history information but is transmitted with the packet.

A value of 1 is shifted into the HIST packet on a taken direct branch (conditional or unconditional) and on a predicate instruction whose predicate condition was evaluated as true. A value of 0 is shifted into the HIST packet on a not taken direct branch and on a predicate instruction whose predicate condition was evaluated as false.

A direct branch is a branch whose target address is hard-coded into the microcode instruction at compile time. An indirect branch is a branch whose target address depends on run time conditions. The indirect branches in the ETPU are return from subroutine, dispatch jump, and dispatch call.

For the ETPU, predicated instructions are those that execute ALU operations conditionally based on the value of the C, Z, or N flags. This conditional execution is controlled by the value of the AS/CE packet in the ETPU microcode instruction formats A3, B4, B5, and B6.



### 53.7.7.3.4 Sequential Instruction Count

The I-CNT packet, present in many of the program trace messages, represents the number of ETPU microcode instructions executed since the start of channel service, last taken direct branch, last not-taken direct branch, last taken indirect branch, or last predicated instruction.<sup>13</sup> In other words, I-CNT is reset whenever it is transmitted within a message or a bit is recorded to HIST.

When a sequential instruction is executed and the current I-CNT value is 255, a resource full message is transmitted to tell the tool that the I-CNT has reached a value of 256. The current HIST packet is transmitted as part of this message and reset inside the NDEDI. The HIST and I-CNT information in the resource full message are combined with information from subsequent messages to provide the development tool a full picture of the program flow.

If additional HIST bits are recorded between a resource full message for the I-CNT and the next program trace message, the I-CNT overflow can be ignored by the tool. In this case, the HIST packet transmitted by any resource full messages and the next program trace are concatenated to provide the full HIST information.

Multiple I-CNT resource full messages can be received between other program trace messages. In these cases, the HIST packets are concatenated, and the I-CNT values are added only when no HIST bits were recorded between the resource full messages.

### 53.7.7.3.5 Interleaved ENGINE1 and ENGINE2 messages

Both ENGINE1 and ENGINE2 can be enabled for program trace at the same time. Each Engine microcode instruction takes two system clocks to execute. The dual Engine system alternates instruction completion between ENGINE1 and ENGINE2, thus one module never completes an instruction in the same clock as the other.

The NDEDI block formats and queues ENGINE1 and ENGINE2 messages in the order they are generated. There is no way for the development tool to infer any other temporal information from the ETPU messages.

## 53.7.7.4 Timing Diagrams

Note that all of the following timing diagrams assume a 3-bit SRC size and either a 4-bit MDO port or a 16-bit MDO port. The state variable is not a signal, but instead is derived from `MSEO`. It is included for clarity. Refer to [Figure 53-23](#) for MSEO state diagram.

The following abbreviations are used for the state variable in the diagrams:

13. ETPU does not have a conditional indirect branch. Thus, there is no not-taken indirect branch.

- ID = Idle
- SM = Start Message
- NT = Normal Transfer
- EP = End Packet
- EM = End Message

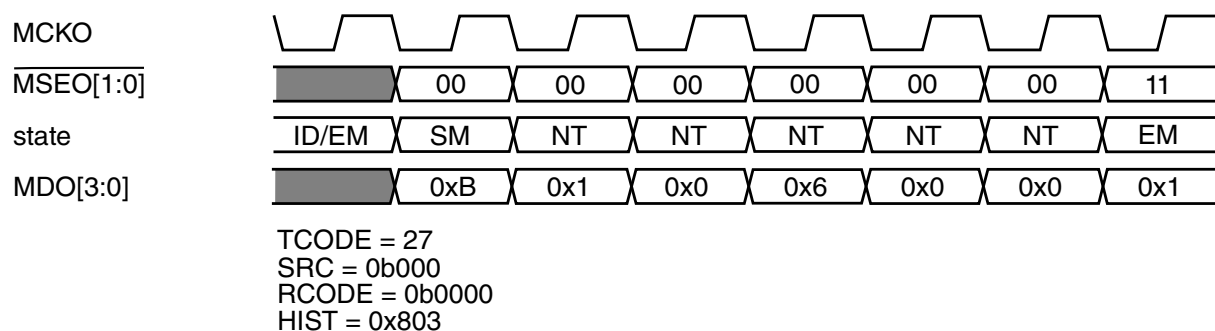


Figure 53-48. Resource Full Message, I-CNT Overflow

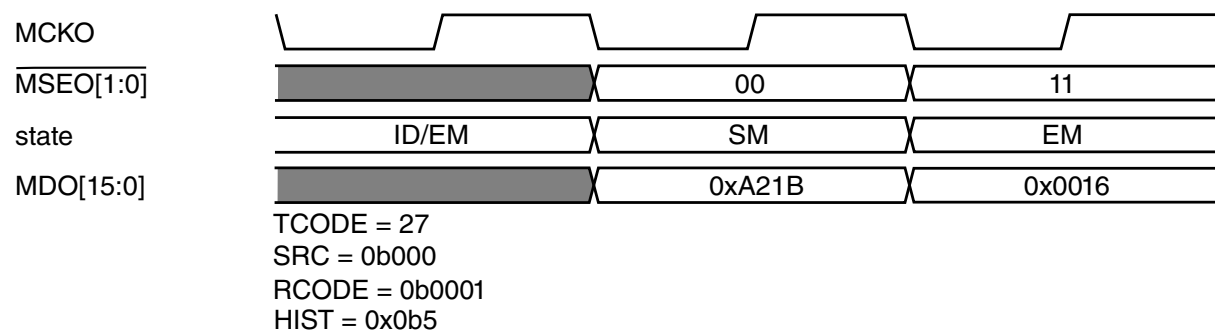


Figure 53-49. Resource Full Message, HIST Buffer Full

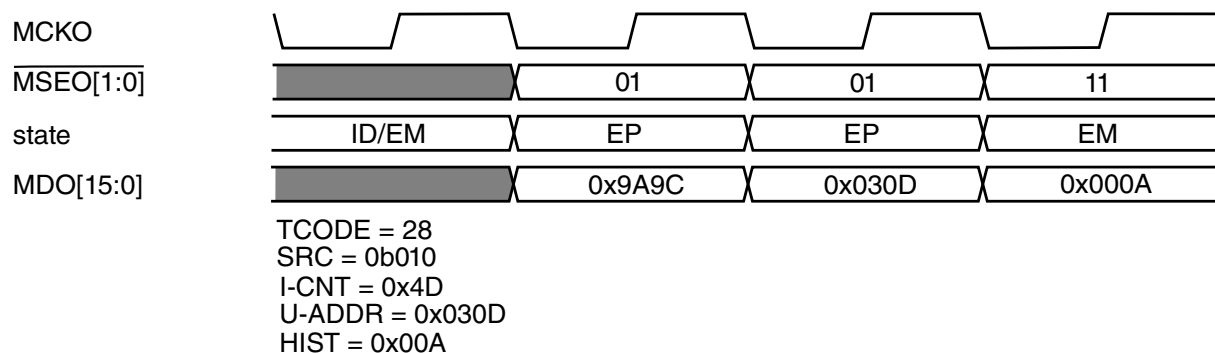
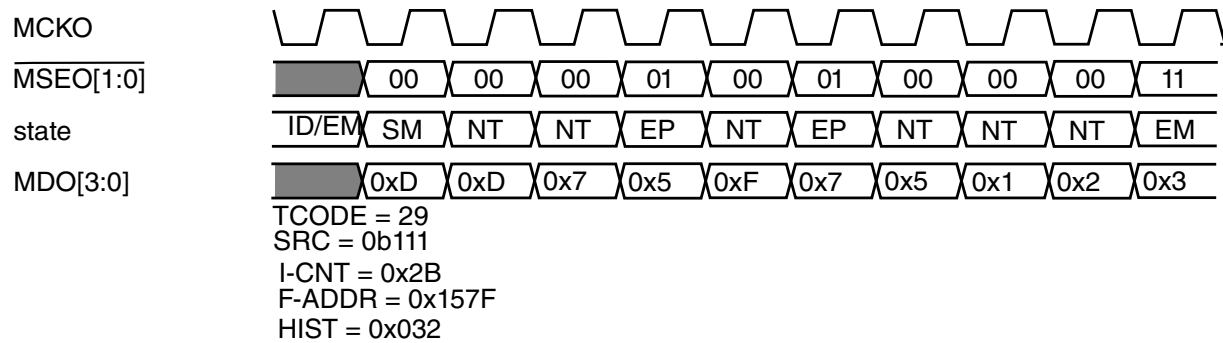
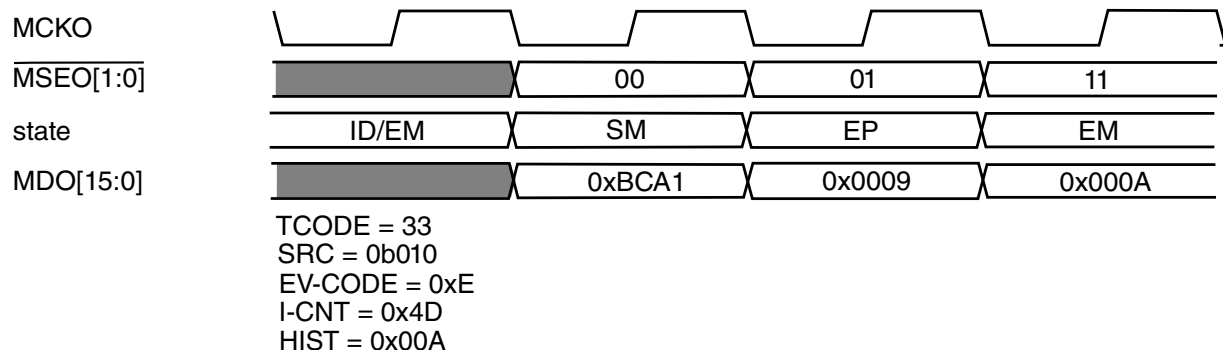
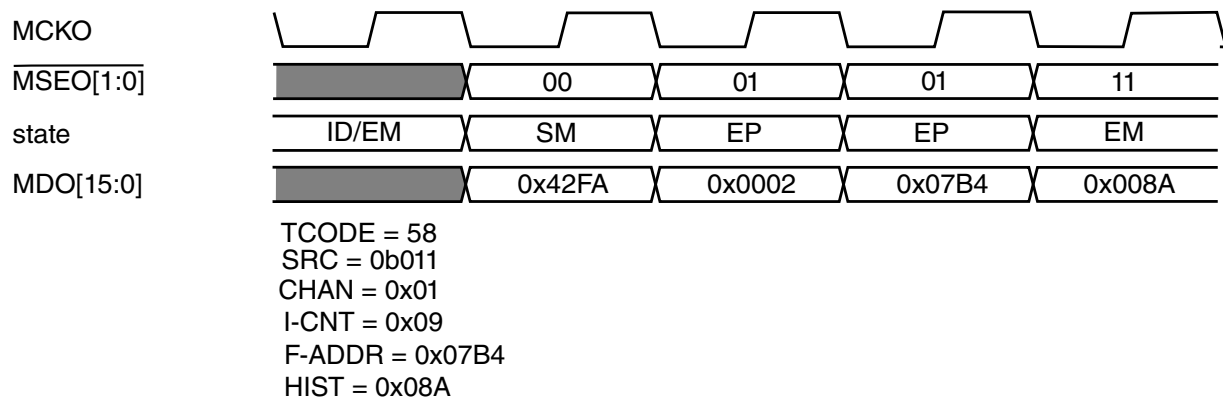
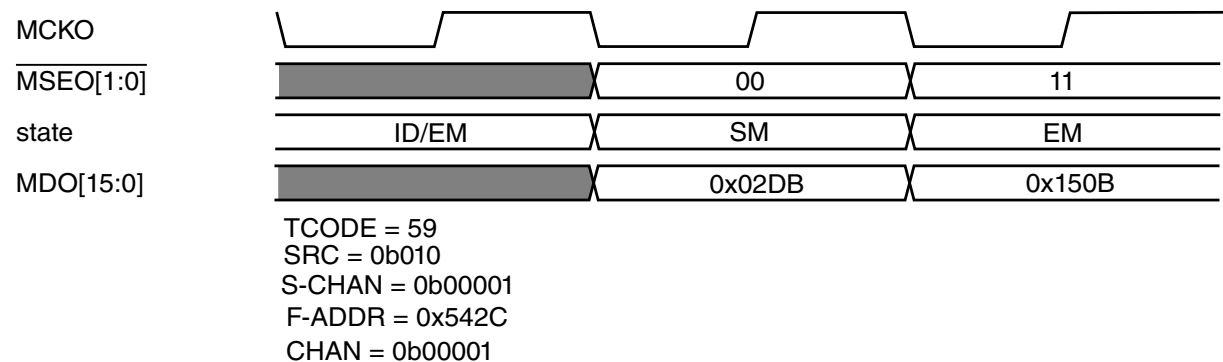


Figure 53-50. Indirect Branch with History Message

**Figure 53-51. Indirect Branch with History Synchronization Message****Figure 53-52. Program Trace Correlation Message****Figure 53-53. Channel Start Service Message****Figure 53-54. Channel Trace Enable Message with the same value for CHAN and S-CHAN**

## Functional description

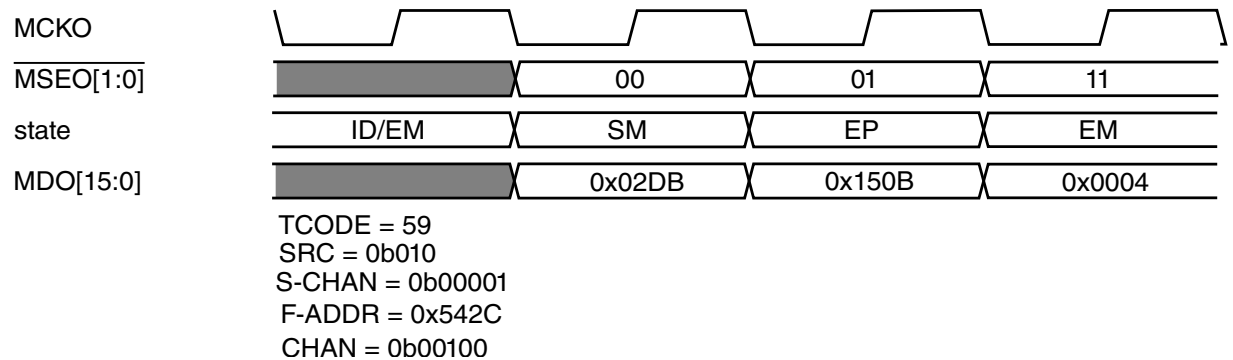


**Figure 53-55. Channel Trace Enable Message with different values for CHAN and S-CHAN**

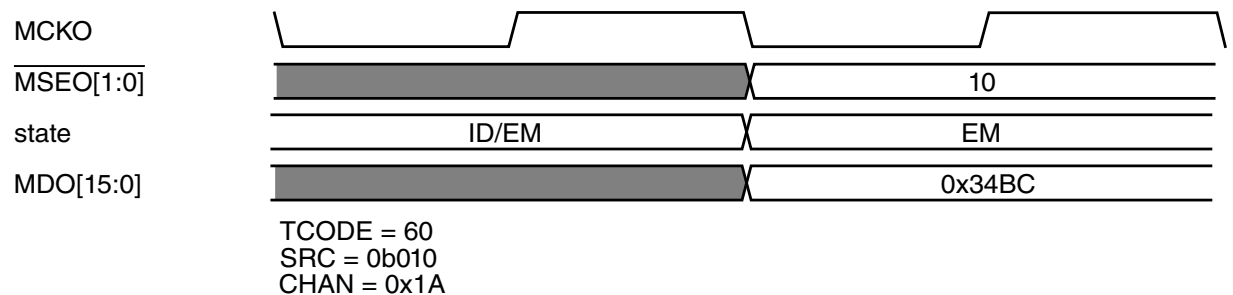


**Figure 53-56. Channel Register Write Message**

## 53.7.8 Data Trace

Dual Engines perform loads and stores to the Shared Parameter RAM (SPRAM). The NDEDI block gathers information about these accesses on a dedicated bus. The NDEDI block traces all SPRAM accesses that meet the selected address range and attributes. This includes accesses from either Engines as well as Coherent Dual-Parameter Controller (CDC) accesses. SPRAM accesses can have data sizes of 8, 24, and 32 bits. The NDEDI block supports all three data sizes.

Whenever performing 8-bits accesses, the ETPU accesses the 8 most significant bits of the word addressed, while the 24-bits accesses are accomplished by accessing the 24 least significant bits. Thus, since the ETPU system uses the Big Endian convention and since the ETPU data addresses are byte-relative the two least significant bits of the transmitted address follows the convention shown in the following table.

**Table 53-33. Two least significant bits for Data Trace Addresses**

| Value | Access                        |
|-------|-------------------------------|
| 0b00  | ETPU 8-bit or 32-bit accesses |
| 0b01  | ETPU 24-bit accesses          |
| 0b1x  | Are not meaningful for ETPU   |

### 53.7.8.1 Data Trace Message Formats

There are four types of Data Trace Messages (DTMs):

- Data Write Message
- Data Read Message
- Data Write with Synchronization Message
- Data Read with Synchronization Message

All of the data trace messages contain a SIZE packet. The following table shows the decoding for this packet.

**Table 53-34. Data Trace SIZE Packet Decodings**

| SIZE Packet | Access Size    |
|-------------|----------------|
| 0b00        | Not Meaningful |
| 0b01        | 24 bits        |
| 0b10        | 8 bits         |
| 0b11        | 32 bits        |

#### 53.7.8.1.1 Data Write Message

The data write message contains the data write value and the address of the target location relative to the address of the previous data trace message from the same source.

The data write message has the format shown in the following figure.

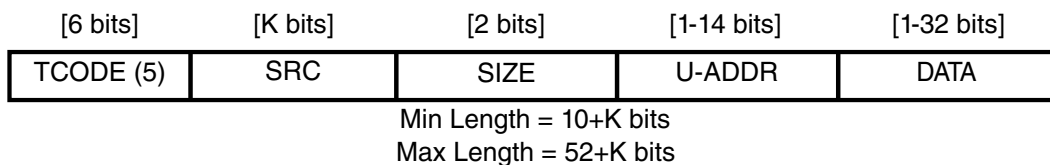


**Figure 53-57. Data Write Message Format**

#### 53.7.8.1.2 Data Read Message

The data read message contains the data read value and the address of the target location relative to the address of the previous data trace message from the same source.

The data read message has the format shown in the following figure.

## Functional description

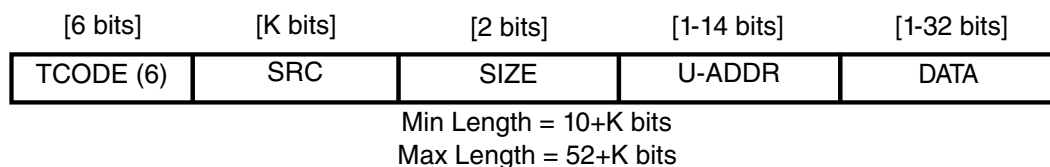


**Figure 53-58. Data Read Message Format**

### 53.7.8.1.3 Data Trace Synchronization Messages

The occurrence of any of the following conditions requires the following data trace message to be synchronizing:

- Exit of system reset
- Exit of low power mode
- Exit of debug mode
- Write to the development control register or data trace control register<sup>14</sup>
- Write to any of the ETPU data trace address range registers
- Exit of NDEDI reset
- 255 data trace messages of the same source are queued without synchronization
- Assertion of the Event In ( $\overline{\text{EVTI}}$ ) pin, depending on how the development control register is configured
- A watchpoint occurrence<sup>15</sup>
- A queue overrun

Data trace synchronization messages provide the full addresses (without leading zeros) and ensure that the development tools fully synchronize with data trace regularly. Each synchronization message provides a reference address for subsequent DTMs, in which only the unique portion of the data trace address is transmitted.

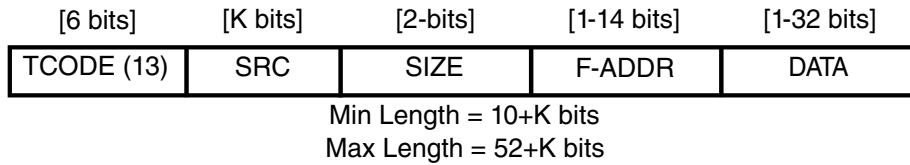
There are two types of data trace synchronization messages: data write synchronization and data read synchronization.

#### 53.7.8.1.3.1 Data Write Synchronization Message

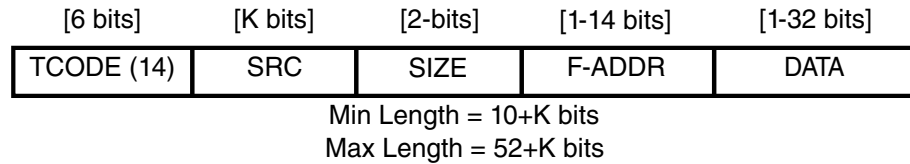
The data write synchronization message has the format shown in the following figure.

14. Only the Source that have the register written will generate a synchronization message.

15. A Watchpoint occurrence for one Engine generates synchronization message for that Engine only.

**Figure 53-59. Data Write Synchronization Message Format****53.7.8.1.3.2 Data Read Synchronization Message**

The data read synchronization message has the format shown in the following figure.

**Figure 53-60. Data Read Synchronization Message Format****53.7.8.2 Data Trace Operation**

Data tracing is performed by snooping a dedicated ETPU/Nexus interface for SPRAM read and write cycles. Data trace functions are enabled by setting the appropriate fields in the following registers:

- ENGINE1 Development Control (NDEDI\_ENGINE1\_DC) Register
- ENGINE2 Development Control (NDEDI\_ENGINE2\_DC) Register
- ENGINE1 Data Trace Control (NDEDI\_ENGINE1\_DTC) Register
- ENGINE2 Data Trace Control (NDEDI\_ENGINE2\_DTC) Register
- CDC Data Trace Control (NDEDI\_CDC\_DTC) Register
- ETPU Data Trace Address Range 0 (NDEDI\_ETPU\_DTAR0) Register
- ETPU Data Trace Address Range 1 (NDEDI\_ETPU\_DTAR1) Register
- ETPU Data Trace Address Range 2 (NDEDI\_ETPU\_DTAR2) Register
- ETPU Data Trace Address Range 3 (NDEDI\_ETPU\_DTAR3) Register

For details on register configuration, refer to [Register descriptions](#).

The dual Engines and CDC data tracing share the same Data Trace Address Range Registers. This enables usage of more address ranges for one client when the others are not being traced, or monitoring a certain range regardless of which client accesses the range.

The ETPU provides information to the NDEDI block when the access is actually being performed. The NDEDI does not have to track a bus protocol or check for error conditions on the accesses, since SPRAM does not generate errors. Any access signaled by the ETPU/NDEDI interface completes without error. Data trace flow is depicted in the following figure.

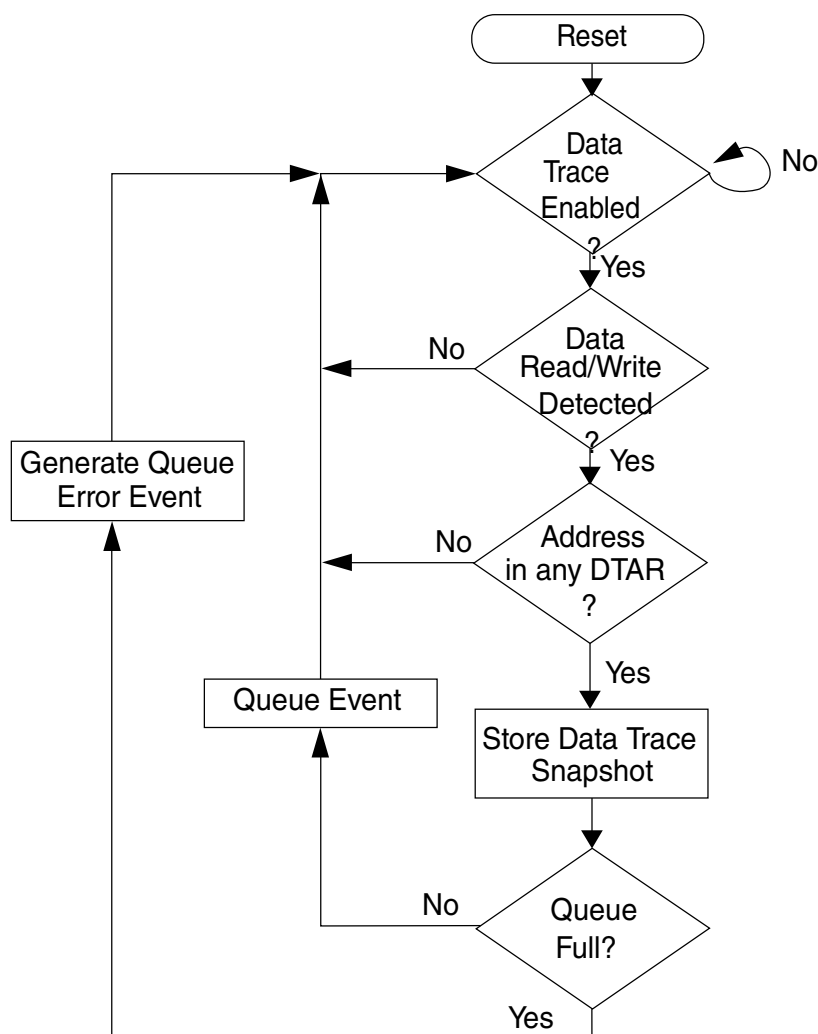


Figure 53-61. ETPU/CDC Data Trace Flow Diagram

### 53.7.8.2.1 Data Trace Windowing

Data trace windowing is provided so the development tool can decrease the auxiliary port usage by limiting the accesses that are traced.



Data trace windowing is achieved via the address range defined by the DTSA and the DTEA fields of the DTAR registers. These registers are shared by ENGINE1, ENGINE2, and the CDC. SPRAM accesses will be traced if their address fall in any of these ranges and the specific range is enabled in the data trace control register of the source making the access. Data read and/or data write trace may be enabled via the Read/Write Trace Control fields in the data trace control registers.

Data trace ranges are 32-bit aligned. This alignment is done by making the two least significant bits of the DTSA and DTEA fields read only. Since the two least significant bits of DTSA and DTEA are read only and default to different values, DTSA can never equal DTEA. The following figure shows the relationship between the DTSA and DTEA fields.

| Programmed Value | Ranged Selected                             |
|------------------|---------------------------------------------|
| DTSA < DTEA      | DTSA → ← DTEA                               |
| DTSA > DTEA      | All addresses are out of Range <sup>1</sup> |
| DTSA == DTEA     | Impossible <sup>2</sup>                     |

**Figure 53-62. Data Trace Address Range Options**

### 53.7.8.2.2 Relative Addressing

The relative address feature is compliant with the IEEE-ISTO 5001-2002 standard recommendations and is designed to reduce the number of bits transmitted for addresses of data read and data write messages.

The address transmitted for a determined source is relative to the address of the previous data trace message sent for that source. It is generated by XORing the new address with the previous address and then using only the results up to the most significant '1' in the result. To recreate this address, an XOR of the (most-significant 0-padded) message address with the previously decoded address gives the current address. The following figure shows how a relative address is generated and how it can be used to recreate the original address.

Previous Address (A1) = 0xC10, New Address (A2) = 0xF64

**Address Generation:**

A1 = 1100 0001 0000  
A2 = 1111 0110 0100  
A1 ^ A2 = 0011 0111 0100  
Address Message (M1) = 11 0111 0100

**Address Recreation:**

A1 ^ M1 = A2  
A1 = 1100 0001 0000  
M1 = 0011 0111 0100  
A1 ^ M1 = 1111 0110 0100 = A2

Figure 53-63. Relative Address Generation and Recreation1

53.7.8.3 Timing Diagrams

Note that all of the following timing diagrams assume a 3-bit SRC size and either a 4-bit MDO port or a 16-bit MDO port. The state variable is not a signal, but instead is derived from  $\overline{\text{MSEO}}$ . It is included for clarity. Refer to [Figure 53-23](#) for  $\overline{\text{MSEO}}$  state diagram.

The following abbreviations are used for the state variable in the diagrams:

- ID = Idle
- SM = Start Message
- NT = Normal Transfer
- EP = End Packet
- EM = End Message

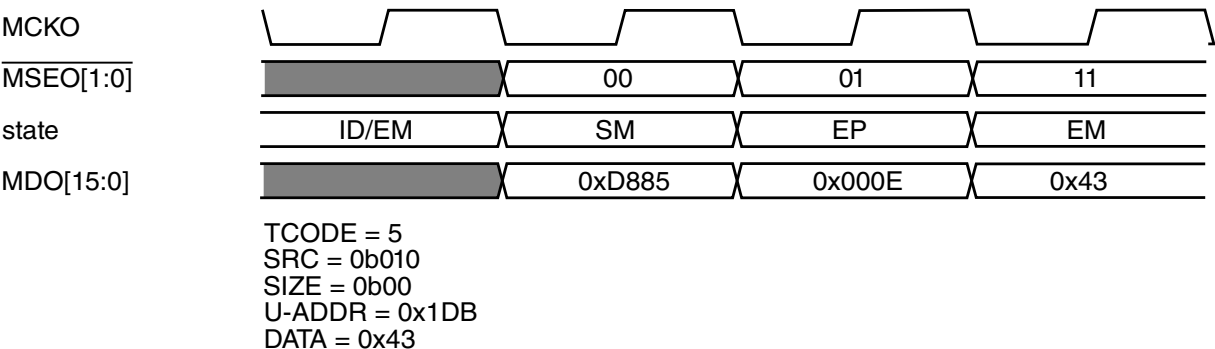


Figure 53-64. Data Write Message

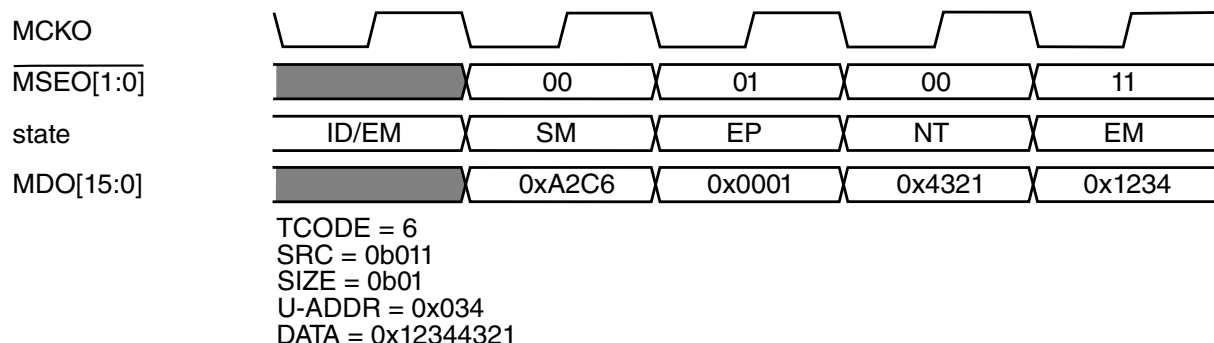


Figure 53-65. Data Read Message

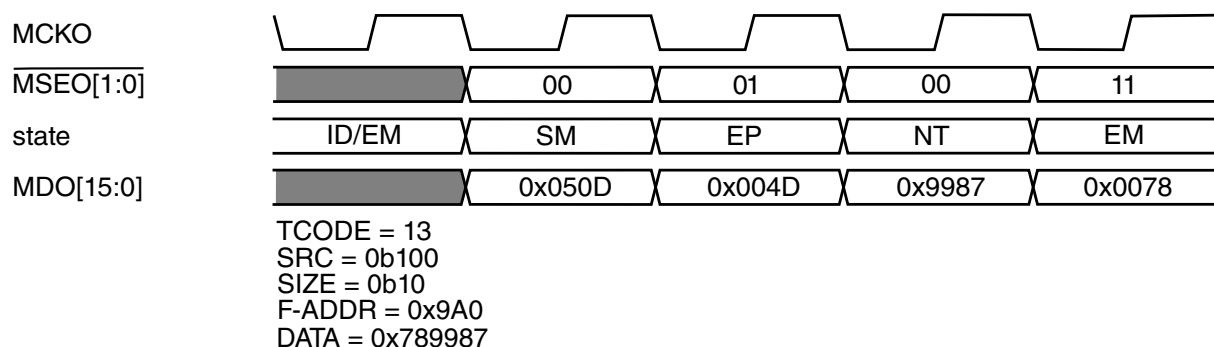


Figure 53-66. Data Write with Synchronization Message

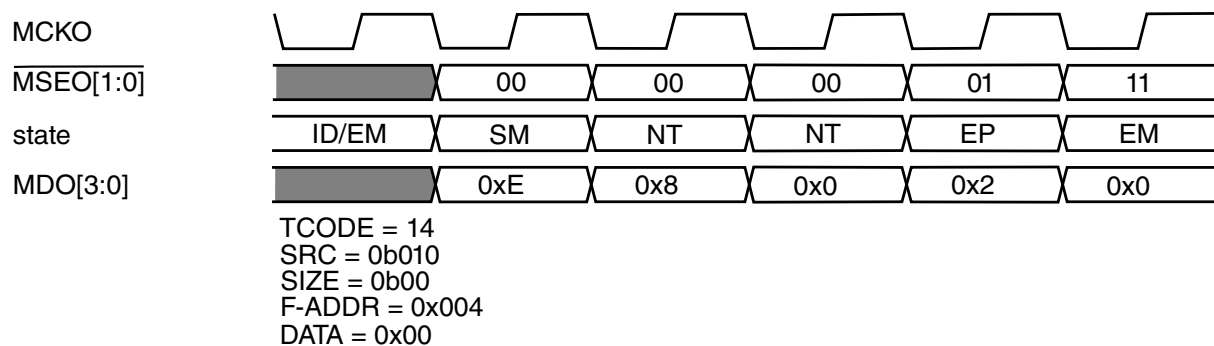


Figure 53-67. Data Read with Synchronization Message

### 53.7.9 Watchpoint Trace

The NDEDI module provides ETPU watchpoint messaging via the auxiliary port, as defined by the IEEE-ISTO 5001-2002 standard.

### 53.7.9.1 Messaging

The NDEDI block provides watchpoint messaging using IEEE-ISTO 5001-2002 standard-defined public messages. When a watchpoint occurs, a watchpoint event is sent to the Event Queue. If the watchpoint condition occurs while the Event Queue is not enabled for storing snapshots, a watchpoint overrun message is generated.

The watchpoint message has the format shown in the following figure.

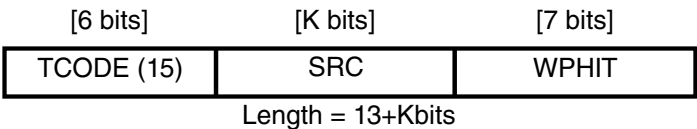


Figure 53-68. Watchpoint Hit Message Format

The values for the WPHIT packet are described in [Table 53-35](#) . Notice that WPHIT is never 0 because the message is only generated when a watchpoint has occurred.

Table 53-35. WPHIT Values

| Value     | Meaning                                    |
|-----------|--------------------------------------------|
| 0b1xxxxxx | ETPU Watchpoint 1 (based on BWC1 register) |
| 0bx1xxxxx | ETPU Watchpoint 2 (based on BWC2 register) |
| 0bxx1xxxx | Channel Register Write Watchpoint          |
| 0bxxx1xxx | Host Service Request Watchpoint            |
| 0bxxxx1xx | Link Register Watchpoint                   |
| 0bxxxxx1x | MRL Watchpoint                             |
| 0bxxxxxx1 | TDL Watchpoint                             |

### 53.7.9.2 Error Messages

A watchpoint overrun error event is queued if an watchpoint event occurs and the Event Queue is not enabled for storing snapshots. This event is queued as soon as the Event Queue gets empty.

The error message has the format shown in the following figure.

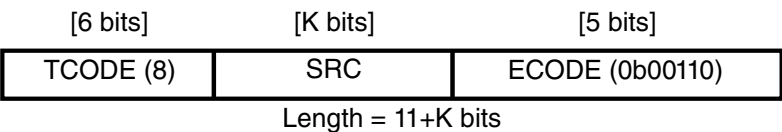


Figure 53-69. Watchpoint Overrun Error Message Format

### 53.7.9.3 Synchronization

Upon the occurrence of a watchpoint, the next program and data trace messages are synchronization messages. A watchpoint in one of the Engines causes data and program synchronization for that Engine only.

### 53.7.9.4 Timing Diagrams

Note that all of the following timing diagrams assume a 3-bit SRC size and a 16-bit MDO port. The state variable is not a signal, but instead is derived from  $\overline{\text{MSEO}}$ . It is included for clarity. Refer to [Figure 53-23](#) for  $\overline{\text{MSEO}}$  state diagram.

The following abbreviations are used for the state variable in the diagrams:

- ID = Idle
- SM = Start Message
- NT = Normal Transfer
- EP = End Packet
- EM = End Message



**Figure 53-70. Watchpoint Message**

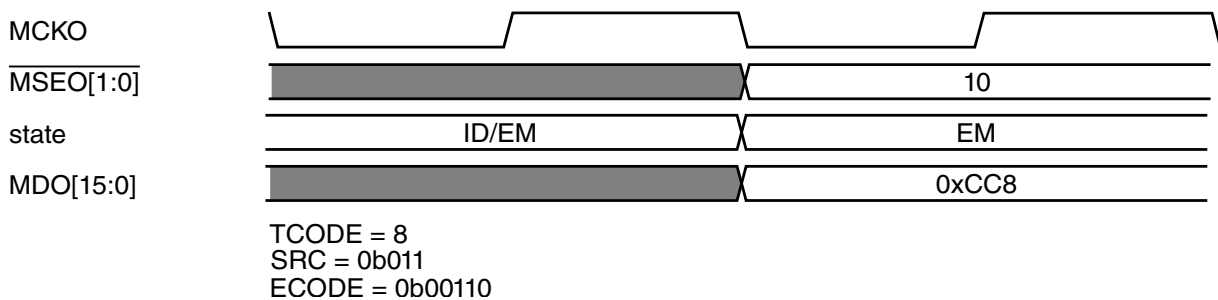


**Figure 53-71. Watchpoint Overrun Error Message**

## 53.7.10 ETPU Message Queue

The NDEDI implements an external Q\_SIZE location Queue that is shared by data trace, ownership trace, program trace, watchpoint, error and debug status events.

One single microcycle may generate a combination of data trace, ownership trace, program trace, watchpoint, error and debug status events. The data trace events related to one engine have different timing though, being generated at the system clock subsequent to the other events. Thus, since the ENGINE1 and ENGINE2 alternate instructions, a snapshot may contain data trace information regarding to one engine and other events information related to the other engine.

With the occurrence of one or more events at one specific cycle, a snapshot of the signals, that can give information about all events, is queued in only one location of the Queue. The information of the source which have generated the events is also queued at the same position, so the Message Formatter can reconstruct the messages of the occurred events. [Figure 53-72](#) shows the Event Queue Block Diagram.

The Queue operation is transparent to the user.

The Event Queue is read whenever there is at least one position of the FIFO occupied and there are no concurrent writes to the Queue. Thus, the appropriate message is formatted and a request to send the message is sent to the Nexus Port Controller (NPC). The request will be assigned a level based on the number of used positions in the Queue as shown in the following table.

**Table 53-36. MDO Request Level**

| Level | Used positions (q) as a fraction of the Queue size |
|-------|----------------------------------------------------|
| 0     | $q = 0$                                            |
| 1     | $1 \leq q < 0.5$                                   |
| 2     | $0.5 \leq q < 0.75$                                |
| 3     | $0.75 \leq q^1$                                    |

1. The MDO request will also be 3 whenever the Queue is disabled for storing new snapshots due to an overrun.

According to the request priority of the NDEDI and the other Nexus Blocks within the chip, the NPC will grant the Auxiliary port to the NDEDI, and as soon as the auxiliary port is granted, the NDEDI will send the message. If there are other pending messages in the same snapshot, a new request is sent and a new grant is expected from the NPC. The next snapshot is only read when there are no pending messages to be sent in the current snapshot and there are no concurrent write accesses to the Queue.

If more than one message event are to be sent, the Message Formatter sends first the Messages related to Debug Status, then the Messages related to Watchpoints, then the messages related to Program Trace and latest the Data Trace Messages.

If the Q\_SIZE position of the Queue is full and an event occurs, the Event Queue is disabled for storing any information. The MDO request level will be 3 until the last snapshot is sent. When the Queue gets empty, an Error event is queued containing information of all lost events. At this point, snapshots may be queued again.<sup>16</sup>

NDEDI keep temporal ordering of messages relative to ENGINE1, ENGINE2 and CDC, but not relative to messages generated by other clients sharing the auxiliary port.

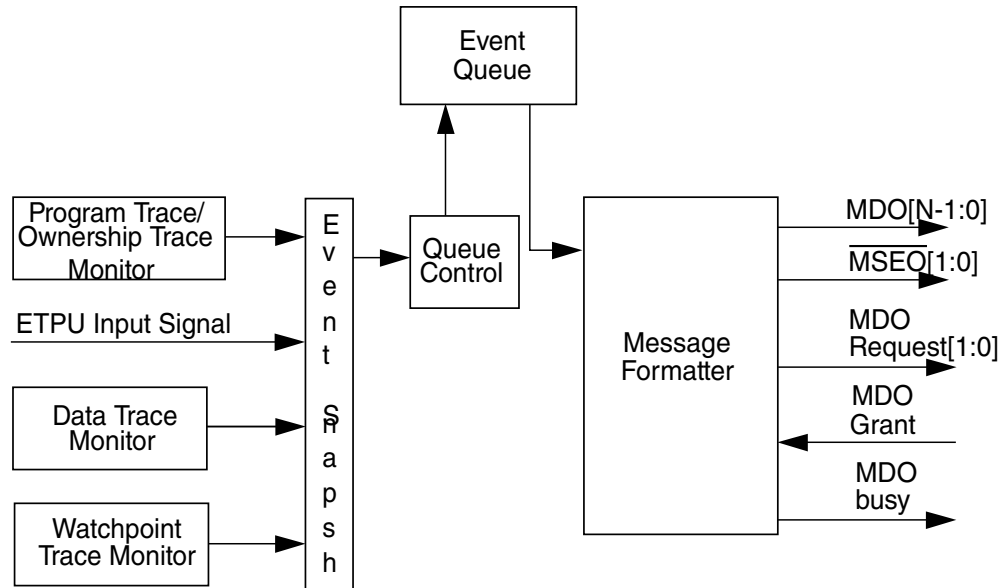


**Figure 53-72. Event Queue**

### 53.7.10.1 Queue Control

The queue is able to perform a single read or write operation per system clock. Thus, since there are no buffers to neither traces nor watchpoint events, the Message Formatter can only read the Event Queue if no events happen at the same time.

### 53.7.10.2 Error Messages

A trace or watchpoint overrun error occurs when an event cannot be queued. The error code (ECODE) within the error message indicates what type of trace overrun has occurred and that messages might have been lost for any or all of these events. [Table 53-25](#) shows the error code encodings for the various overrun error conditions.

16. If ENGINE<sub>x</sub> is in stall condition, and an error happens, the ENGINE<sub>x</sub> operation will only be resumed after the Queue is empty.

When information is lost, the Event Queue is disabled and no events are queued until the Event Queue gets empty. After the Event Queue gets empty an error message is queued indicating which information was lost for each source. At this point the Queue is enabled for storing snapshots again.

If multiple sources lose information at the same time multiple error messages are sent. In this case, the message related to the Engine 1 will be sent first, then the message related to the Engine 2 and finally the Error message related to the CDC.

The error message has the format shown in the following figure.

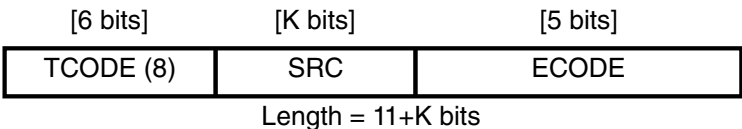


Figure 53-73. Error Message Format

53.7.10.3 Timing Diagrams

Note that the following timing diagram assumes a 3-bit SRC size and a 16-bit MDO port. The state variable is not a signal, but instead is derived from  $\overline{\text{MSEO}}$ . It is included for clarity. Refer to [Figure 53-23](#) for  $\overline{\text{MSEO}}$  state diagram.

The following abbreviations are the values for the state variable in the diagram:

- ID = Idle
- SM = Start Message
- NT = Normal Transfer
- EP = End Packet
- EM = End Message

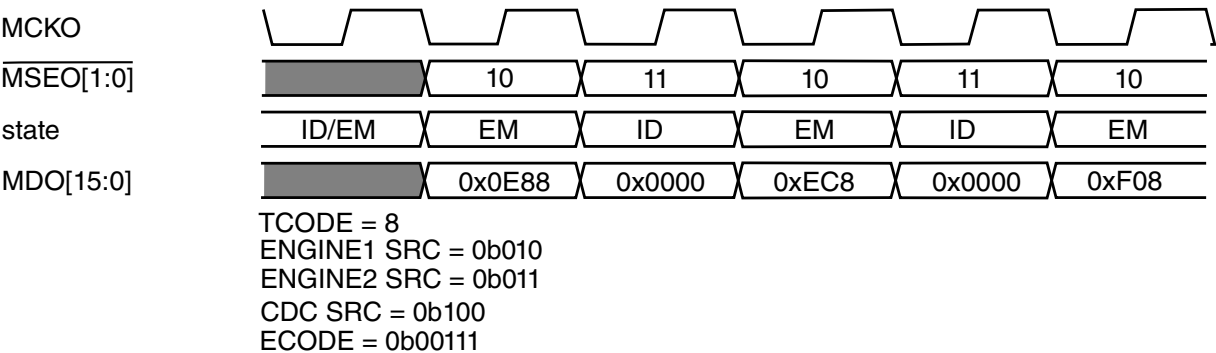


Figure 53-74. Error Messages (Program/Data/Ownership Trace Overrun)



Notice that three overrun messages are transmitted in the preceding figure. The first is an overrun indication for ENGINE1, the second is an overrun indication for ENGINE2, and the third is an overrun indication for the CDC.

## 53.8 Initialization/Application Information

The following section describes the Initialization/Application Information process.

### 53.8.1 Accessing NDEDI tool-mapped registers

To initialize the JTAG port for register accesses, the following sequence is required:

1. Enable the Nexus TAP controller
2. Retrieve Device ID if needed
3. Load the Nexus TAP controller with the NEXUS-ENABLE instruction

To write control data to NDEDI tool-mapped registers, the following sequence is required:

1. Write to the client select control register, if needed
2. Write the 7-bit register index and set the write bit to select register with a pass through the SELECT-DR-SCAN path in the JTAG state machine
3. Write the register value with a pass through the SELECT-DR-SCAN path. Notice that the prior value of this register is shifted out during the write

To read status and control data from NDEDI tool-mapped registers, the following sequence is required:

1. Write to the client select control register if needed
2. Write the 7-bit register index and clear write bit to select register with a pass through SELECT-DR-SCAN path
3. Read the register value with a pass through the SELECT-DR-SCAN path. Data shifted in is ignored

See [IEEE 1149.1 \(JTAG\) Input Port](#) for more detail.

## 53.8.2 Program Trace Reconstruction

This section details the program trace reconstruction mechanism supported by Nexus3 for the NDEDI. Program trace is implemented via Branch Trace Messaging (BTM) as per the Class 3 IEEE-ISTO 5001-2002 standard.

### 53.8.2.1 Branch Trace Messages

Branch Trace Messaging allows program trace by providing the following types of information:

- Messaging for start of program trace
  - Channel Start Message (CSM)
  - Channel Trace Enable Message (CTM)
- Branches with run-time target address (indirect branches)
  - Indirect Branch with History Message (IHM)
  - Indirect Branch with History and Synchronization Message (IHS)
- Discontinuity Information:
  - History including information of predicated instruction execution and information of all branches taken or not taken with compile-time target address. (direct branches)
  - Information of the amount of sequential instructions executed since the last branch. (ICNT)
    - Channel Start Message (CSM)
    - Indirect Branch with History and Synchronization Message (IHS)
    - Resource Full Message (RFM)
    - Program Correlation Message (PCM)
- Information of Program Trace End
  - Channel Start Message (CSM)
  - Program Correlation Message (PCM)
- Program trace overrun

- An Error Message will be issued if program trace information is lost due to FIFO overrun. The following error codes indicates program trace error: ECODE = 0b00001, 0b00111, 0b01000, 0b11001 and 0b11010

### 53.8.2.2 Assembly Code

The assembly code is necessary for gathering information about the direct branches location, target addresses for direct branches, predicated instructions location and "non-flush" indication.

The following eTPU instruction formats<sup>17</sup> can generate direct branches:

E1, E2, E3 and E4

The following eTPU instruction formats can be predicated:

A3, B4, B5 and B6 but only if AS/CE differs 0, 1 and 7.

The following eTPU instruction formats can be indirect branches:

A1 with RTN = 1, D1, D2, D3 and D4

The Branch instructions have the capability of executing one instruction before changing the flow to the target address. The instruction being executed before the change of flow is hereafter called "non flushed" instruction.

The following eTPU instruction formats can lead to "non flushed" instructions:

D1, D2, D3, D4, E1, E2, E3 and E4 if FLS = 1

The other formats will lead to sequential instructions.

### 53.8.2.3 Branch Indication

The branch indication within the eTPU/NDEDI system happens at the instruction execution instead of the branch target execution. Thus it is important to notice that the "non flushed" instruction will be stored in the subsequent HIST/ICNT information. It is also important to notice that a CTM message is not issued if program trace is enabled in a "non flushed" instruction. In this case a CTM will be issued in the subsequent instruction.

17. Please refer to the ETPU Block Guide for further information about the instructions format.

### 53.8.2.4 Code Reconstruction

The CSM and CTM messages indicate the starting program trace address. Thus, these messages must be used as a starting point in order to synchronize the reconstruction program with the Assembly code.

From the starting point, the program flow can be completely reconstructed unless an Error message is issued. If an Error message is issued, watchpoints must be used to enable/disable program trace reducing the number of program trace messages.

### 53.8.2.5 Program Reconstruction Examples

Follows a code example where the reconstruction can be performed with the generated messages.

The program reconstruction will be performed by using the following two components:

- A copy of the program. Ideally this would be a listing file from the assembler;
- The program trace messages sent to the development tool;

#### 53.8.2.5.1 Example 1

Follows the Example Code file:

```
234 FFFFFFFF format:F1.
235 010F64D1 format:A2 t2abd:"RAR", imm:578.
236 F7E846FF format:E4 j/c:"jump",bcc:"false",fls:"no_flush",bcf:"true",
    baf:567.
237 8AD4A218 format:B1 shf:"ror",cin:"no_carry",binv:"inv",t2abd:"P[31:24]",
    rw:"write",t4abs:"TRR",t4bbs:"BS=0,1st_2nd",src:">>1",p/d:"P",
    ccs:"flag_size",end:"yes",aid:24.
```

Follows the messages generated for a certain run example:

1. CSM (s=0x3) [S-CHAN=0x1e] [I-CNT=0x0000] [faddr=0x0234] [HIST=0x00000001];
2. PCM (s=0x3) [event=0x0] [I-CNT=0x01] [HIST=0x00000001];
3. CTM (s=0x3) [S-CHAN=0x001e] [faddr=0x0235] [CHAN=0x00000000];
4. PCM (s=0x3) [event=0x0] [I-CNT=0x00] [HIST=0x00000002];
5. CTM (s=0x3) [S-CHAN=0x001e] [faddr=0x0237] [CHAN=0x00000000];
6. CSM (s=0x3) [S-CHAN=0x11] [I-CNT=0x0001] [faddr=0x0443] [HIST=0x00000001];

## Reconstruction:

The reconstruction starts with CSM pointing to the 0x0234 address. Note that ICNT = 0x0 and HIST = 0x1 indicating that there was no previous thread being executed.

PCM with event = 0x0 indicates that eTPU entered in debug mode. ICNT = 0x01 and HIST = 0x1 means that there were no direct branches and only one sequential instruction. Thus, instruction at address 0x0234 was executed and then eTPU entered in debug mode.

The third message is CTM which indicates that Program Trace was enabled again (eTPU exited debug mode) at address 0x0235. Then, another PCM with event = 0x0 was issued indicating entry in debug mode. From the code it is possible to inquire that instruction at address 0x0235 is a sequential instruction and the instruction at address 0x0236 is a conditional jump with "no flush".

HIST = 0b10 indicates that the first branch after address 0x0235 was reached but not taken. ICNT = 0x0 indicates that no sequential instruction after the last branch was executed, thus, it is possible to conclude that the code entered in debug mode just after executing the instruction at address 0x0236.

The fifth message is CTM which indicates a new Program Trace enable at address 0x0237.

The next message refers to CSM, and it indicates that a new thread is starting at address 0x0443. HIST=0b1 and ICNT=0b1 refers to the old thread and states that one sequential instruction (at address 0x0237) was executed before starting the new thread. This is the expected behavior since the address 0x0237 contains a "end=yes" command.

### 53.8.2.5.2 Example 2

Follows the Example Code file:

```
336 CB5CE283 3083 format:D1 mrle:"match_off",psc:"high",fls:"flush",rw:"write",
    pscs:"opac2",flc:"flags=P[25:24]",circ:"nop",
    r/d:"dispatch_call",p/d:"P",rsiz:"24",aid:131,
    end:"no".
...
41B 00AF6531 3962 format:A2 t2abd:"RAR", imm:1065.
41C F32083BF 3965 format:E4 j/c:"jump",bcc:"LSR",fls:"no_flush",bcf:"false",
    baf:1053.
41D 11092F98 3968 format:A1 t2d:"DIOB",rtn:"nop",IMM:9608768.
41E 004F7757 3971 format:B3 shf:"<<1",cin:"use_carry",binv:"inv",t4bbs:"SR",rw:"read",
    t4abs:"AS[23:0]=0",t2abd:"MACH",src:">>1",p/d:"DIOB",
    ccs:"no_flags",zro:"clear",end:"yes",
    stc:"no_inc/dec".
```

Follows the messages generated for a certain run example:

1. CSM (s=0x3) [S-CHAN=0x0f] [I-CNT=0x0001] [faddr=0x0336]  
[HIST=0x0000000a];

2. IHM (s=0x3) [I-CNT=0x00] [uaddr=0x072d] [HIST=0x00000001];
3. CSM (s=0x3) [S-CHAN=0x14] [I-CNT=0x0003] [faddr=0x0345] [HIST=0x00000003];

#### Reconstruction:

The first CSM indicates that the thread will be started at the address 0x0336. The next message is an IHM with I-CNT = 0x0 and HIST = 0x1 indicating that no instructions were executed until the indirect branch. The message indicates u-addr = 0x072d, thus the target address of this instruction is 0x041b (0x0336 XOR 0x072d). Note that since the instruction at 0x0336 is a flush instruction, then the instruction at 0x0337 is not supposed to be executed.

The next message refers to CSM, and it indicates that a new thread is starting at address 0x0345. Since HIST = 0x3, either the next branch (after address 0x041b) was taken or the next predicated instruction (after address 0x041b) was executed. From address 0x041b the next branch instruction is located at 0x041c address, thus all instructions were executed until this address.

The instruction within the address 0x041c refers to a branch with no\_flush instruction to address 0x041d.

Since I-CNT = 0x3, it is expected that 3 sequential instructions are executed after the branch instruction.

The instruction within 0x041d will be executed since it is a non-flushed instruction, then it will be executed again since 0x041d is the target for the branch instruction executed before, and finally the instruction within the address 0x041e is executed. Note that the instruction within 0x041e contains a "end=yes" command which will finish the current thread.

### 53.8.3 Microcode Development Support

The following sections describe the steps required to perform various class 1 features with the NDEDI block.

#### 53.8.3.1 Read and Write SPRAM In Debug Mode

Reading and writing to the SPRAM is done through Nexus read/write access hardware external to the NDEDI block. Refer to the MCUs SoC Guide for details. SPRAM reads and writes are done the same in debug mode as they are in run mode.

### 53.8.3.2 Read and Write ETPU Internal Registers in Debug Mode

1. Select an area of the SPRAM to be used for dumping the register values
2. Read the current value of those SPRAM locations and store them
3. Write to the SPRAM as needed for writing internal ETPU registers
4. Execute external microcode that reads values from the SPRAM location and writes the values to registers or reads registers and writes the values to the SPRAM. See [Execute Forced Microcode Instruction in Debug Mode](#) for more details
5. Read the SPRAM as needed to retrieve register read values
6. Write the SPRAM back to the original values

### 53.8.3.3 Enter Debug Mode at the Negation of Reset

If the DBE field of the ETPU's Development Control register is asserted, NDEDI ETPU will enter debug mode as soon as the system reset is negated. Configure other bits as appropriate.

### 53.8.3.4 Enter Debug Mode During Normal Execution

To enter debug mode during normal execution, write the ETPU's Development Control register to set the DBE and DBR bits. Configure other bits as appropriate.

### 53.8.3.5 Stop Program Execution on a Breakpoint

There are four possibilities to stop the program execution on a Breakpoint:

1. Program execution halts at the end of the current microcycle when a synchronous breakpoint condition is reached. A synchronous breakpoint condition occurs due to the following breakpoints:
  - Breakpoint due to DBR assertion
  - Twin Engine breakpoint<sup>18</sup>

---

18. Depending on CBT value

- External breakpoint<sup>19</sup>
  - Breakpoint due to exit of system reset
  - breakpoint due to start of a channel service
  - Single Step breakpoint
  - breakpoint due to EVTI assertion
  - breakpoint due to MISC miscompare
2. Program execution stops at the beginning of the current microcycle when an asynchronous breakpoint condition is reached. An asynchronous condition occurs due to the following breakpoints:
    - breakpoint due to SPRAM access
    - breakpoint due to Illegal instruction
    - breakpoint due to CHAN register write
  3. Program execution stops at the end of the current thread if a twin breakpoint or an external source requests to enter in debug mode and the NDEDI is programmed to enter in debug mode only at the end of the current thread
  4. Whenever fetching an instruction, if the microprogram counter matches the BWA related to one of the breakpoint conditions programmed for breaking into an instruction fetch, the instruction is tagged indicating that it must not be executed. The microengine halts just before the execution of this instruction if it was supposed to be executed<sup>20</sup>.

See [Set Breakpoint or Watchpoints](#) and [Register descriptions](#) for details on configuring breakpoints.

#### 53.8.3.5.1 Multiple breakpoint conditions at the same address

The ETPU block must always execute the first instruction after exiting debug mode.

Assume that an engine is halted due to a synchronous breakpoint at a given address. As soon as the engine exits debug mode, the instruction at the given address will be executed anyway, even if other breakpoint conditions are matched, for instance tagged instructions or asynchronous breakpoints.

---

19. Depending on CBT value

20. Instructions flushed after a JUMP are not considered to be executed



Although the DS register doesn't signalize the other breakpoint conditions, the development tool can infer from the NDEDI registers and pre-loaded code if the engine is halted in one of these conditions.

### 53.8.3.6 Single Step Instructions and Re-Enter Debug Mode

To single step an instruction from shared code memory in debug mode, write to the Engine's Development Control register so the CBR bit is asserted, the DBR bit is cleared and the SS bit is asserted. Single stepping causes the Engine to exit debug mode for a microcycle where the TCRs, decrementors, and microprogram counter can update. Program Trace messages are generated within Single Step operation.

### 53.8.3.7 Set Breakpoint or Watchpoints

There are two main ETPU breakpoint/watchpoint sources that are independent of each other but have identical configuration registers. Each can be configured as either a breakpoint or watchpoint that detects reads and/or writes, compares data and/or address. Each can be configured to occur only on certain Channel services.

In addition, there is a breakpoint/watchpoint that can be configured to occur when the CHAN register is written to a certain value.

There are also four channel service breakpoint/watchpoints that can be configured to occur when a specific channel is serviced for one of the following reasons: host service request, link request, match detect, and transition detect.

Breakpoints cause the Engine to halt. Watchpoints cause a watchpoint hit message. All breakpoint/watchpoint sources can be configured to assert EVTO on an breakpoint/watchpoint occurrence.

Refer to [Register descriptions](#) for detailed descriptions of breakpoint/watchpoint registers.

### 53.8.3.8 Execute Forced Microcode Instruction in Debug Mode

To execute a forced instruction in debug mode, write the instruction into the Microinstruction debug register. This mechanism does not cause the Engine to exit debug mode so the microprogram counter is not updated (unless a branch instruction is executed) and the TCRs and decrementors are not updated if the CLKS bit is asserted in the Development Control Register. Executing instructions in this manner is useful for reading and writing internal Engine registers.



## Chapter 54

# Development Trigger Semaphore (DTS)

### 54.1 Introduction

The Development Trigger Semaphore (DTS) module enables software to signal an external tool by driving a persistent (affected only by reset or an external tool) signal on an external device pin. There are a variety of ways this module can be used, including as a component of an external real-time data acquisition system.

#### Note

When used as a component of a triggered data acquisition system, Nexus read/write access is different than the data acquisition protocol defined in the IEEE-ISTO 5001-2003 or IEEE-ISTO 5001-2012 Nexus standards, which use the Nexus Auxiliary port.

### 54.2 Overview

The Development Trigger Semaphore (DTS) module consists of registers and a small amount of combinational logic to generate an output signal—DTS Trigger Output (DTO). The registers are as follows:

- **DTS\_SEMAPHORE** register—Any bit in this 32-bit register, when set to a value of logic '1', causes the DTS module output signal to be asserted if the corresponding **DTS\_EN** bit is set in the **DTS\_ENABLE** register. This enables an external tool to detect up to 32 signals from the application software. In an application, each bit is generally associated with a specific data set. Only the processor core and DMA module can set bits in this register. The bits can only be cleared by a tool access via Nexus Read/Write Access.

- **DTS\_SEMAPHORE\_B** register—Any bit in this 32-bit register, when set to a value of logic '1', causes the DTS module output signal to be asserted if the corresponding **DTS\_EN\_B** bit is set in the **DTS\_ENABLE** register. This enables an external tool to detect an additional 32 signals from the application software. In an application, each bit is generally associated with a specific data set. Only the processor core and DMA module can set bits in this register. The bits can only be cleared by a tool access via Nexus Read/Write Access.
- **DTS\_STARTUP** register—This register provides a mechanism for the external tool to notify software running on the CPU that the tool is connected and can provide information about either the type of tool or options that can be used by the software.
- **DTS\_ENABLE** register—This register provides an enable/disable capability for the DTS feature.

The architecture is shown in the following figure.

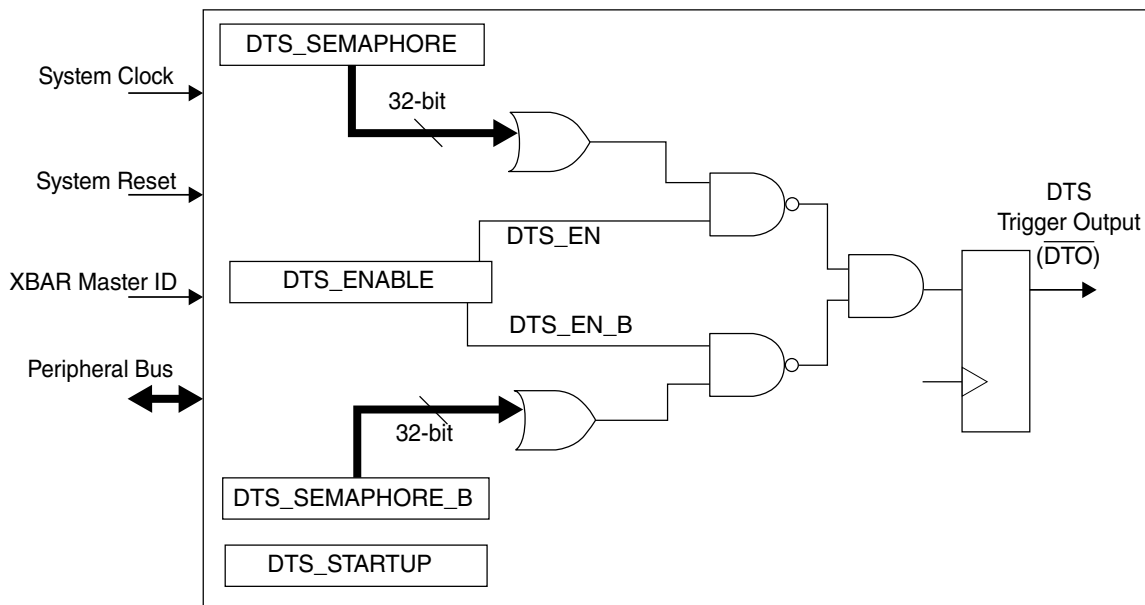


**Figure 54-1. DTS block diagram**

The DTS Trigger Output (DTO) signal is connected to one of the EVTO inputs of the Nexus Port Controller (NPC). The other EVTO inputs are connected to the other Nexus modules in the device. DTO is asserted when any bit in a semaphore register is set and the DTS function of that register is enabled.

### Note

When the DTS module is enabled ( $\text{DTS\_ENABLE}[\text{DTS\_EN}] = 0b1$  or  $\text{DTS\_ENABLE}[\text{DTS\_EN\_B}] = 0b1$ ), all other Nexus EVTO functions associated should be disabled by the tool and

EVTO becomes the DTO. Unlike the EVTO function that only asserts for one clock, the DTO function remains asserted until the tool reads the semaphore register that is the source of the assertion, clearing the register's contents.

The following figure shows an example of the chain of events that begins with setting of a bit in the DTS\_SEMAPHORE register and the clearing of the register caused by a Nexus read.

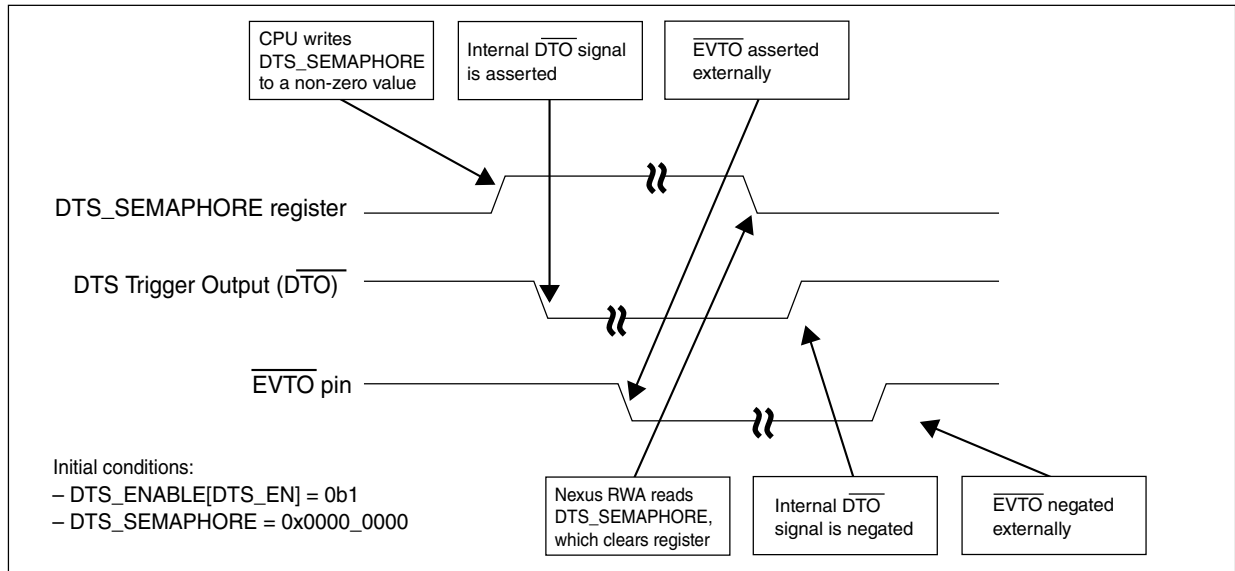


Figure 54-2. DTO event sequence example

### 54.3 DTS device connections

The following figure shows the DTS device connections. The DTS module connects to the Peripheral Bridge (PBRIDGE) for access to the registers. The PBRIDGE is connected to the device modules through a slave port of the Crossbar bus interface (XBAR).

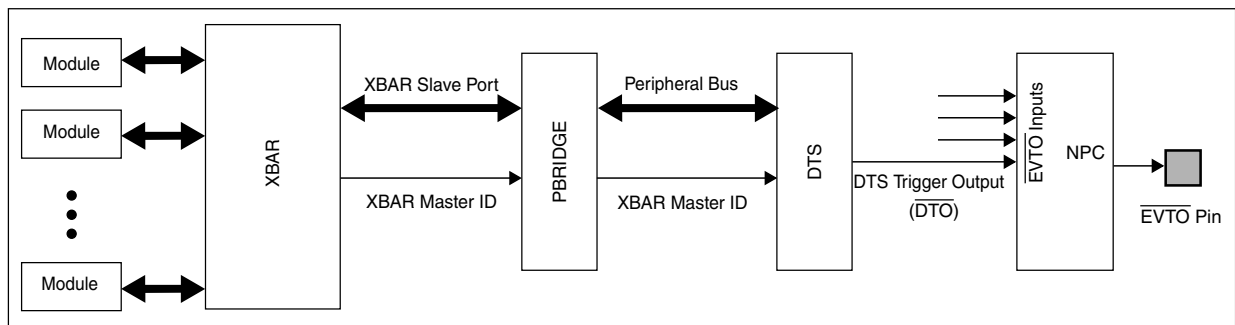


Figure 54-3. DTS device connections

The registers have limited access as described in [DTS register access](#). Access is based on the XBAR Master ID of the accessing module. Writing to the semaphore registers is limited to the cores and the eDMA module and is restricted to only setting bits. Only an access via a Nexus Read/Write Access from an external tool can clear bits in the semaphore registers (semaphore register bits are cleared automatically when read). Similarly, the DTS\_ENABLE and DTS\_STARTUP registers can only be written via a Nexus Read/Write Access.

### Note

Nexus Read/Write Accesses use the core load/store bus to perform accesses, but Nexus accesses have a different Master ID than normal core load/stores.

## 54.3.1 DTS register access

A summary of accesses to all DTS registers by bus masters is provided in the following table. Only proper 32-bit accesses are valid. The effects of write accesses that are not 32 bits are not defined.

**Table 54-1. DTS register access effects**

| Register        | 32-bit read                 |      |      |                        | 32-bit write     |           |           |                        |
|-----------------|-----------------------------|------|------|------------------------|------------------|-----------|-----------|------------------------|
|                 | RWA <sup>1</sup>            | Core | eDMA | Other crossbar masters | RWA <sup>1</sup> | Core      | eDMA      | Other crossbar masters |
| DTS_ENABLE      | Data                        | Data | Data | Data                   | Data             | No effect | No effect | No effect              |
| DTS_STARTUP     | Data                        | Data | Data | Data                   | Data             | No effect | No effect | No effect              |
| DTS_SEMAPHORE   | Data and Clear <sup>2</sup> | Data | Data | Data                   | No effect        | Bit OR    | Bit OR    | No effect              |
| DTS_SEMAPHORE_B | Data and Clear <sup>2</sup> | Data | Data | Data                   | No effect        | Bit OR    | Bit OR    | No effect              |

1. Nexus Read/Write access via an external tool

2. A read of the semaphore registers by Nexus Read/Write Access module is destructive and clears all bits in the register

Access to DTS module registers is controlled based on the XBAR Master ID of the accessing module.

### Note

The XBAR Master ID should not be confused with the Master Port number of the XBAR. See the Crossbar Switch (XBAR) chapter for details.

Tools must access the DTS registers through the Nexus Read/Write Access mechanism of a core. External tool accesses through either core appear as if the access is via the core and therefore do not have the same level of access as a Nexus Read/Write Access.

## 54.4 Memory mapped registers

Only certain types of accesses are allowed to the DTS registers. See the description for each register to determine the accesses that are allowed.

**DTS memory map**

| Address offset (hex) | Register name                         | Width (in bits) | Access | Reset value | Section/ page               |
|----------------------|---------------------------------------|-----------------|--------|-------------|-----------------------------|
| 0                    | Output Enable Register (DTS_ENABLE)   | 32              | R/W    | 0000_0000h  | <a href="#">54.4.1/2483</a> |
| 4                    | Startup Register (DTS_STARTUP)        | 32              | R/W    | 0000_0000h  | <a href="#">54.4.2/2484</a> |
| 8                    | Semaphore Register (DTS_SEMAPHORE)    | 32              | R/W    | FFFF_FFFFh  | <a href="#">54.4.3/2485</a> |
| C                    | Semaphore Extension (DTS_SEMAPHORE_B) | 32              | R/W    | FFFF_FFFFh  | <a href="#">54.4.4/2486</a> |

### 54.4.1 Output Enable Register (DTS\_ENABLE)

The DTS\_ENABLE register controls the DTS Trigger Output ( $\overline{D\bar{T}O}$ ) and whether  $\overline{D\bar{T}O}$  is active on the  $\overline{E\bar{V}\bar{T}O}$  output pin of the device. The DTS\_ENABLE register can be read by the core, but can only be written by a Nexus Read Write Access (RWA).

#### NOTE

Access to the DTS\_SEMAPHORE, DTS\_SEMAPHORE\_B, and DTS\_STARTUP registers are unaffected by the state of this register.

Address: 0h base + 0h offset = 0h

| Bit   | 0        | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|-------|----------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| R     | Reserved |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |          |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

## Memory mapped registers

|       |          |    |    |    |    |    |    |    |    |    |    |    |    |    |          |        |
|-------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----------|--------|
| Bit   | 16       | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30       | 31     |
| R     | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    | DTS_EN_B | DTS_EN |
| W     |          |    |    |    |    |    |    |    |    |    |    |    |    |    |          |        |
| Reset | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0        | 0      |

### DTS\_ENABLE field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–29<br>Reserved | This field is reserved.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 30<br>DTS_EN_B   | DTS Enable B. Controls whether the $\overline{D\bar{T}O}$ signal is routed to the $\overline{E\bar{V}T\bar{O}}$ pin. The DTS Enable bit is cleared by a device reset (either the assertion of the external $\overline{R\bar{E}S\bar{E}T}$ or by an internally generated reset). A Nexus reset does not change the state of this register.<br><br>0 DTS output disabled. Any bit set in the DTS_SEMAPHORE_B register does not assert the DTS trigger output signal.<br>1 DTS output enabled. Any bit set in the DTS_SEMAPHORE_B register asserts the DTS trigger output signal ( $\overline{D\bar{T}O}$ ). |
| 31<br>DTS_EN     | DTS Enable. Controls whether the $\overline{D\bar{T}O}$ signal is routed to the $\overline{E\bar{V}T\bar{O}}$ pin. The DTS Enable bit is cleared by a device reset (either the assertion of the external $\overline{R\bar{E}S\bar{E}T}$ or by an internally generated reset). A Nexus reset does not change the state of this register.<br><br>0 DTS output disabled. Any bit set in the DTS_SEMAPHORE register does not assert the DTS trigger output signal.<br>1 DTS output enabled. Any bit set in the DTS_SEMAPHORE register asserts the DTS trigger output signal ( $\overline{D\bar{T}O}$ ).       |

## 54.4.2 Startup Register (DTS\_STARTUP)

The DTS\_STARTUP register is used for tool detection and startup information exchange between external data acquisition tool and the embedded controller. DTS\_STARTUP register can be read by the core, the eDMA module and Nexus but can only be updated by a Nexus Read Write Access (RWA).

Address: 0h base + 4h offset = 4h

|       |    |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | AD |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |    |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

### DTS\_STARTUP field descriptions

| Field      | Description                                                                                                                                                                                                         |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>AD | Application Dependent register bits. The bits have no defined meaning to the microcontroller. They are used to by an external tool to pass information (for example, application options and status) to application |



**DTS\_STARTUP field descriptions (continued)**

| Field | Description                                                                                                                                                                                                                                                                                                                                  |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | software running on target microcontroller at startup time. Use a Nexus RWA 32-bit write access to update the contents of this register.<br><br>A device reset (either from the $\overline{\text{RESET}}$ pin or an internally generated reset) clears all bits in the register. A Nexus reset does not change the contents of the register. |

**54.4.3 Semaphore Register (DTS\_SEMAPHORE)**

The DTS\_SEMAPHORE register is used by software to assert the  $\overline{\text{DTO}}$  signal on the device  $\overline{\text{EVTO}}$  pin. A one in any bit of this register causes the  $\overline{\text{DTO}}$  signal on the  $\overline{\text{EVTO}}$  pin to be driven low. The intended use of this register is for the  $\overline{\text{DTO}}$  signal to notify tools that data is available. Individual bits are used to identify the specific data.

Address: 0h base + 8h offset = 8h

|       |               |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0             | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | <div>ST</div> |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |               |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 1             | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  |    |

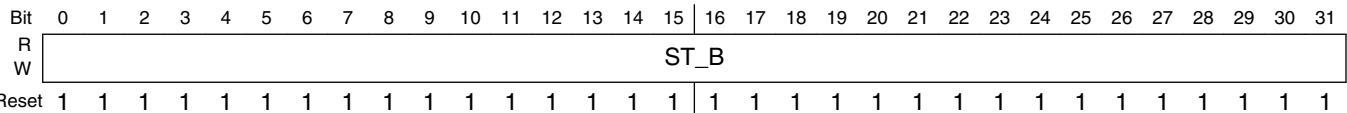
**DTS\_SEMAPHORE field descriptions**

| Field      | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>ST | <p>Semaphore Trigger. When a core or eDMA writes a logical '1' to a bit, the bit is set. A write of '0' by the core or DMA does not change the state of the bit.</p> <ul style="list-style-type: none"> <li>All register bits are set to '1' by a device reset.</li> <li>A Nexus reset does not change the state of this register.</li> <li>The register can be accessed, with restrictions, by any core, DMA or any Nexus RWA.</li> <li>For the core or DMA, only 32-bit write or read accesses are valid.</li> <li>A core or DMA valid read access returns the current value of the register and leaves the register unchanged.</li> </ul> <p>0 No flag<br/>1 Flag is set</p> |

54.4.4 Semaphore Extension (DTS\_SEMAPHORE\_B)

The DTS\_SEMAPHORE\_B register is used by software to assert the  $\overline{DTO}$  signal on the device  $\overline{EVTO}$  pin. A one in any bit of this register causes the  $\overline{DTO}$  signal on the  $\overline{EVTO}$  pin to be driven low. The intended use of this register is for the  $\overline{DTO}$  signal to notify tools that data is available. Individual bits are used to identify the specific data.

Address: 0h base + Ch offset = Ch



DTS\_SEMAPHORE\_B field descriptions

| Field        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>ST_B | <div>Semaphore Trigger Extension. When a core or eDMA writes a logical '1' to a bit, the bit is set. A write of '0' by the core or DMA does not change the state of the bit.<ul style="list-style-type: none"><li>• All register bits are set to '1' by a device reset.</li><li>• A Nexus reset does not change the state of this register.</li><li>• The register can be accessed, with restrictions, by any core, DMA or any Nexus RWA.</li><li>• For the core or DMA, only 32-bit write or read accesses are valid.</li><li>• A core or DMA valid read access returns the current value of the register and leaves the register unchanged.</li></ul></div> <div>0 No flag</div> <div>1 Flag is set</div> |

54.5 Example application

The calibration process of a new engine requires real-time access to calibration tables and the ability to update the tables in real-time The DTS module enables this capability by enabling software to assert a signal to an external device pin to notify an external tool that data is available. The tool can then retrieve the data.

In this type of application the DTS\_SEMAPHORE register and DTS Trigger Output (DTO) signal provide a mechanism to notify the calibration tool that the calibration variable or variables (or sets of measurements), up to 32, have been updated with new values and are available for the tool to access.

## Note

It is the user's responsibility to ensure that the tool has time to retrieve the data prior to that particular trigger being set a second time. It is also permissible to have multiple triggers active at the same time or for a second trigger to be set before a previous trigger has been serviced, as long as it is not the same trigger (unless it is acceptable to the tool to not receive every data set).

The following table shows an example DTS startup sequence for an external real-time data acquisition system. The startup and synchronization sequence can be as simple or as complicated as the need requires. However, a typical startup sequence is as follows:

1. The DTS\_STARTUP register is cleared by a power on reset or any CPU reset.
2. The tool writes a non-zero value to the DTS\_STARTUP register.
3. The CPU (user application software) then reads the value of the DTS\_STARTUP register. Based on this value, different initialization options can be selected. The bits can be used for any application specific definitions.
4. Since the DTS\_SEMAPHORE register is cleared when the tool reads the current value. The tool should perform all necessary initialization before reading this register. The application software can then check that the DTS\_SEMAPHORE register was cleared by the tool, to determine that it is safe to start using it for its intended raster trigger semaphore function.
5. An optional hand shake from the CPU can be used to inform the tool that the user software has detected that the tool is attached and the CPU has performed the proper initialization for the tool by writing a predefined value to the DTS\_SEMAPHORE register (the example shown in the figure above uses 0xAAAA\_AAAA—all A's was used since it is unrealistic that 16 channels could be enabled very quickly after start up after a reset).

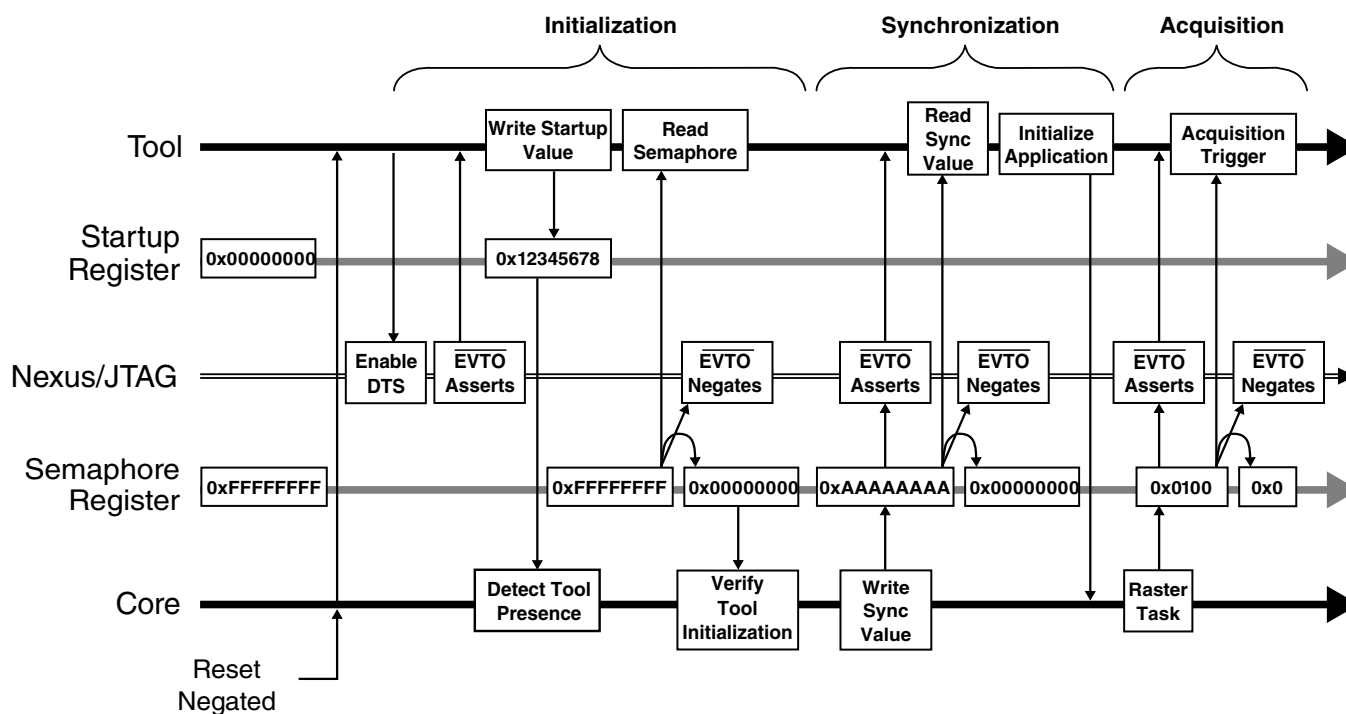


Figure 54-8. DTS startup sequence example

# Chapter 55

## Fault Collection and Control Unit (FCCU)

### 55.1 Chip-specific FCCU information

The FCCU on this chip supports 46 non-critical faults (NCFs).

#### 55.1.1 FCCU register default values

The following table documents FCCU register default values that are specific to this chip. For the default values of all other FCCU registers, see [Register descriptions](#).

**Table 55-1. FCCU register default values**

| Register       | Default value           |
|----------------|-------------------------|
| FCCU_CFG       | 0000_0000h <sup>1</sup> |
| FCCU_NCF_CFG0  | FFFF_FFFFh              |
| FCCU_NCF_CFG1  | 0000_FFFFh              |
| FCCU_NCFS_CFG0 | 2200_2000h              |
| FCCU_NCFS_CFG1 | 0000_0000h              |
| FCCU_NCFS_CFG2 | 0000_0000h              |
| FCCU_NCF_E0    | 0000_5040h              |
| FCCU_NCF_E1    | 0000_0000h              |
| FCCU_NCF_TOE0  | 0000_0000h              |
| FCCU_NCF_TOE1  | 0000_0000h              |
| FCCU_NCF_TO    | 0000_FFFFh              |
| FCCU_CFG_TO    | 0000_0006h              |

1. FCCU\_CFG resets to 0000\_0000h after the chip resets the FCCU. After a Configuration-state timeout, the value of FCCU\_CFG is 0000\_0008h. For details, including the value of FCCU\_CFG after an OP31 operation, see [FCCU\\_CFG register event bit values by source \(N and C\)](#).

## 55.1.2 FOSU\_COUNT value

The FCCU Output Supervision Unit (FOSU) has a timer whose duration is the value FOSU\_COUNT. On this chip, the value is 8000 IRCOSC clock cycles.

## 55.1.3 FCCU fault inputs

The following table identifies the various fault input signals to the FCCU, how they can be tested, and the default reaction to each input after the FCCU itself is reset.

**Table 55-2. FCCU non-critical fault (NCF) mapping**

| NCF | Fault                 | Fault description                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Error reaction path check <sup>1</sup>                                                                  | Default reaction after reset |
|-----|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|------------------------------|
| 0   | TEMP_ERROR            | Event indication from temperature sensor. The fault is cleared by clearing the status in PMC_ESR_TD, followed by clearing the FCCU channel status, FCCU_NCF_Sn[NCFSm].                                                                                                                                                                                                                                                                                                         | LBIST + TSNS_FAULT bit of Temperature Sensor Fault Injection Register (PMC_TS_FIR)                      | No reaction                  |
| 1   | LVD_ERROR             | Voltage out of range indication from LVDs. The fault is cleared by clearing the status in PMC_LVD_HVD_EVENT_STATUS, followed by clearing the FCCU channel status, FCCU_NCF_Sn[NCFSm].                                                                                                                                                                                                                                                                                          | LVD assertion test by tap point manipulation + LVD_FAULT bit of FCCU Fault Injection Register (PMC_FIR) | No reaction                  |
| 2   | HVD_ERROR             | Voltage out of range indication from HVDs. The fault is cleared by clearing the status in PMC_LVD_HVD_EVENT_STATUS, followed by clearing the FCCU channel status, FCCU_NCF_Sn[NCFSm].                                                                                                                                                                                                                                                                                          | HVD assertion test by tap point manipulation + HVD_FAULT bit of FCCU Fault Injection Register (PMC_FIR) | No reaction                  |
| 3   | DPMC_SELF_TEST        | Digital PMC initialization error during self test. The fault is cleared by clearing the FCCU channel status, FCCU_NCF_Sn[NCFSm]. The status gets cleared if the fault is not persistent.                                                                                                                                                                                                                                                                                       | LBIST+ STEST_FAULT bit of FCCU Fault Injection Register (PMC_FIR)                                       | No reaction                  |
| 4   | MEMORY_DCF_SAFETY_ERR | Indication of fault during timing/repair configuration initialization by SSCM of different memories. Faults can be caused by parity or triple-redundant errors in loaded configuration data. The status is cleared by clearing SSCM_STATUS[CERS], followed by clearing the FCCU channel status, FCCU_NCF_Sn[NCFSm]. The status gets cleared if the fault is not persistent. Additionally, this error will cause a MBIST failure. This channel bit is cleared in case of reset. | LBIST                                                                                                   | No reaction                  |
| 5   | SSCM_XFER_ERR         | SSCM transfer or parity error during DCF loading (trimming, STCU, and UTEST Miscellaneous <sup>2</sup> ). This fault is also asserted when there is a flash memory ECC double bit detection during flash memory reset reads or coherency checks done on the flash memory test row reads. The fault is                                                                                                                                                                          | Not testable (but wrong configuration can trigger other detection)                                      | No reaction                  |

*Table continues on the next page...*

**Table 55-2. FCCU non-critical fault (NCF) mapping (continued)**

| NCF | Fault                           | Fault description                                                                                                                                                                                                     | Error reaction path check <sup>1</sup>                                     | Default reaction after reset |
|-----|---------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------|------------------------------|
|     |                                 | cleared by clearing SSCM_STATUS[CER], followed by clearing the FCCU channel status, FCCU_NCF_Sn[NCFSm]. This channel bit is cleared in case of reset.                                                                 |                                                                            |                              |
| 6   | STCU_UF                         | STCU unrecoverable error during self-test. Fault is cleared by clearing STCU2_ERR_STAT[UFSF] followed by clearing the FCCU channel status, FCCU_NCF_Sn[NCFSm].                                                        | SW check of BIST MISR or Feedback loop via FCCU Fake bit                   | Reset                        |
| 7   | STCU_RF                         | STCU recoverable error during self-test. Fault is cleared by clearing STCU2_ERR_STAT[RFSF] followed by clearing the FCCU channel status, FCCU_NCF_Sn[NCFSm].                                                          | SW check of BIST MISR or Feedback loop via FCCU Fake bit                   | No reaction                  |
| 8   | STCU_LMBIST_USR_ERR             | Activation of LBIST or MBIST control during application mode. Fault is cleared by clearing the FCCU channel status, FCCU_NCF_Sn[NCFSm]. The status will clear if the fault is not persistent.                         | Feedback loop via FCCU Fake bit.                                           | No reaction                  |
| 9   | SPURIOUS_DEB_UG_SSCM_ACTIVATION | JTAG, NPC, or Debug functionality moved out of reset or SSCM activation during application mode. Fault is cleared by clearing the FCCU channel status, FCCU_NCF_Sn[NCFSm].                                            | LBIST + Feedback loop via FCCU Fake bit                                    | No reaction                  |
| 10  | RCCU_0                          | Safety cores out of sync during lockstep. Fault is cleared by clearing the FCCU channel status, FCCU_NCF_Sn[NCFSm].                                                                                                   | LBIST + Feedback loop via FCCU Fake bit connected to additional RCCU input | No reaction                  |
| 11  | RCCU_1                          | Safety cores out of sync during lockstep. Fault is cleared by clearing the FCCU channel status, FCCU_NCF_Sn[NCFSm].                                                                                                   | LBIST + Feedback loop via FCCU Fake bit connected to additional RCCU input | No reaction                  |
| 12  | WDRS_DBRC0                      | Core 0 Watchdog Timer reset status (TSR[WRS]) and Core 0 Debug Reset Control (DBCR0[RST]). The fault is cleared by clearing FCCU_NCF_Sn[NCFSm] after the status has been cleared in TSR[WRS].                         | LBIST + SW test by not triggering WD during startup                        | Reset                        |
| 13  | WDRS_DBRC1                      | Core 1 Watchdog Timer reset status (TSR[WRS]) and Core 1 Debug Reset Control (DBCR0[RST]). The fault is cleared by clearing FCCU_NCF_Sn[NCFSm] after the status has been cleared in TSR[WRS].                         | LBIST + SW test by not triggering WD during startup                        | No reaction                  |
| 14  | SWTRS0                          | Software Watchdog A reset. The fault is cleared by clearing FCCU_NCF_Sn[NCFSm] after the status has been cleared inside SWT_A.                                                                                        | LBIST + SW test by not triggering SWT during startup                       | Reset                        |
| 15  | SWTRS1                          | Software Watchdog B reset. The fault is cleared by clearing FCCU_NCF_Sn[NCFSm] after the status has been cleared inside SWT_B.                                                                                        | LBIST + SW test by not triggering SWT during startup                       | No reaction                  |
| 16  | SWTINT0                         | Software Watchdog A interrupt request. The fault is cleared by clearing FCCU_NCF_Sn[NCFSm] after the status has been cleared inside SWT_A.                                                                            | LBIST + SW test by not triggering SWT during startup                       | No reaction                  |
| 17  | FLASH_INIT <sup>3</sup>         | Flash memory initiation error. This fault is intended for debug purposes only because, if such an error occurs, no code will run from flash memory. It is expected that this fault is read through development tools. | LBIST                                                                      | No reaction                  |

*Table continues on the next page...*

**Table 55-2. FCCU non-critical fault (NCF) mapping (continued)**

| NCF | Fault     | Fault description                                                                                                                                                                                                                                                                                                                                               | Error reaction path check <sup>1</sup>                                            | Default reaction after reset |
|-----|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|------------------------------|
| 18  | SSCM_INIT | SSCM initiation error. DCFs might not have loaded properly.                                                                                                                                                                                                                                                                                                     | LBIST                                                                             | No reaction                  |
| 19  | PER_CE    | Indication of correctable errors in eTPU, M_CAN, and FlexCAN RAM Controllers. The fault is cleared by clearing the FCCU channel status, FCCU_NCF_Sn[NCFSm]. For proper eTPU SCM and SDM monitoring, eTPU_MESR_MECR[MECIE] must be set to 1.                                                                                                                     | Error injection in eTPU, FlexCAN, M_CAN + LBIST + Feedback loop via FCCU Fake bit | No reaction                  |
| 20  | PER_UCE   | Indication of uncorrectable errors in eTPU, M_CAN, and FlexCAN RAM Controllers. The fault is cleared by clearing the FCCU channel status, FCCU_NCF_Sn[NCFSm]. The eTPU combines uncorrectable errors and other sources in a single exception signal. The exact exception source must be determined with the help of the eTPU Global Exception.                  | Error injection in eTPU, FlexCAN, M_CAN + LBIST + Feedback loop via FCCU Fake bit | No reaction                  |
| 21  | FLS_CE    | Indication of correctable errors in flash memory. The fault is cleared by clearing the FCCU channel status, FCCU_NCF_Sn[NCFSm].                                                                                                                                                                                                                                 | LBIST                                                                             | No reaction                  |
| 22  | FLS_UCE   | Indication of uncorrectable errors in flash memory. The fault is cleared by clearing the FCCU channel status, FCCU_NCF_Sn[NCFSm].                                                                                                                                                                                                                               | LBIST                                                                             | No reaction                  |
| 23  | PLL_0     | PLL0 loss of lock. For this fault source to be asserted, the Loss of Lock Interrupt must be enabled by writing 1 to PLLDIG_PLL0CR[LOLIE]. Fault is cleared by clearing PLLDIG_PLL0SR[LOLF], followed by clearing FCCU_NCF_Sn[NCFSm].<br><br><b>NOTE:</b> Always configure this fault to generate a reset instead of an interrupt.                               | LBIST + Forced loss by PLL reconfiguration                                        | No reaction                  |
| 24  | PLL_1     | PLL1 loss of lock. For this fault source to be asserted, the Loss of Lock Interrupt must be enabled by writing 1 to PLLDIG_PLL1CR[LOLIE]. Fault is cleared by clearing PLLDIG_PLL1SR[LOLF], followed by clearing FCCU_NCF_Sn[NCFSm].<br><br><b>NOTE:</b> When PLL1 is used as a system clock: Configure this fault to generate a reset instead of an interrupt. | LBIST + Forced loss by PLL reconfiguration                                        | No reaction                  |
| 25  | CMU_0_OSC | Loss of XOSC clock. The fault is cleared by clearing CMU_ISR[OLRI] of CMU_0, followed by clearing the FCCU channel status, FCCU_NCF_Sn[NCFSm].<br><br><b>NOTE:</b> When XOSC is used as a system clock: Configure this fault to generate a reset instead of an interrupt.                                                                                       | LBIST                                                                             | No reaction                  |
| 26  | CMU_0_PLL | Exception from monitor CMU_0 (PLL0). The fault is cleared by clearing CMU_ISR[FHHI], CMU_ISR[FLLI], or both of CMU_0, followed by clearing the FCCU channel status, FCCU_NCF_Sn[NCFSm].                                                                                                                                                                         | LBIST + Test by misconfiguration                                                  | No reaction                  |
| 27  | CMU_COMP  | Exception from monitors CMU_1, CMU_2, and CMU_3. The fault is cleared by clearing CMU_ISR[FHHI], CMU_ISR[FLLI], or both of CMU_n, followed by clearing the FCCU channel status, FCCU_NCF_Sn[NCFSm].                                                                                                                                                             | LBIST + Test by misconfiguration                                                  | No reaction                  |

*Table continues on the next page...*



**Table 55-2. FCCU non-critical fault (NCF) mapping (continued)**

| NCF | Fault                                           | Fault description                                                                                                                                                                                                                     | Error reaction path check <sup>1</sup>                                            | Default reaction after reset |
|-----|-------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|------------------------------|
| 28  | CMU_OTHER                                       | Exception from monitors CMU_4 to CMU_8. The fault is cleared by clearing CMU_ISR[FHHI], CMU_ISR[FLLI], or both of CMU_n, followed by clearing the FCCU channel status, FCCU_NCF_Sn[NCFSm].                                            | LBIST + Test by misconfiguration                                                  | No reaction                  |
| 29  | EDC_ECC_FLASH                                   | Indication of a hardware fault in the ECC of the flash memory resulting in corrupted ECC detection/correction. Fault is cleared by a dummy read of the flash memory followed by clearing the FCCU channel status, FCCU_NCF_Sn[NCFSm]. | LBIST                                                                             | No reaction                  |
| 30  | EDC_ECC_FLASH_C                                 | Indication of hardware fault in the Flash Memory Controller resulting in corrupted ECC detection/correction. Fault is cleared by clearing the FCCU channel status, FCCU_NCF_Sn[NCFSm].                                                | LBIST                                                                             | No reaction                  |
| 31  | ENC_ERR_FLASH                                   | Indication of hardware fault resulting in corrupted flash memory access. Fault is cleared by clearing the FCCU channel status, FCCU_NCF_Sn[NCFSm].                                                                                    | LBIST                                                                             | No reaction                  |
| 32  | ADDR_FDBK_ERROR_FLASH_C                         | Indication of an addressing error in the Flash Memory Controller resulting in corrupted flash memory access. Fault is cleared by clearing the FCCU channel status, FCCU_NCF_Sn[NCFSm].                                                | LBIST                                                                             | No reaction                  |
| 33  | XBAR Transaction Monitor                        | Indication by the Crossbar Integrity Checker (XBIC) of a corrupted transaction in the crossbar. Fault is cleared by clearing the FCCU channel status, FCCU_NCF_Sn[NCFSm].                                                             | LBIST + proper configuration of XBIC Error Injection Register (XBIC_EIR)          | No reaction                  |
| 34  | ADDR_FDBK_ERROR_PRAM + RAM_LWB_ERR <sup>4</sup> | Indication of an addressing error in system RAM or a write error in the RAM Controller resulting in corrupted RAM access. Fault is cleared by clearing the FCCU channel status, FCCU_NCF_Sn[NCFSm].                                   | LBIST                                                                             | No reaction                  |
| 35  | SSCM_PASS_ERROR_CC_ERR                          | An ECC error occurred while SSCM or PASS was fetching data from flash memory.                                                                                                                                                         | LBIST                                                                             | No reaction                  |
| 36  | EDC_ERR_PRAM                                    | Indication of a fault in the RMW logic of the system RAM resulting in an incorrect RMW access and a potentially corrupted RAM word. Fault is cleared by clearing the FCCU channel status, FCCU_NCF_Sn[NCFSm].                         | LBIST                                                                             | No reaction                  |
| 37  | DFT1                                            | Test circuitry Group 1 activation. Fault is cleared by clearing the FCCU channel status, FCCU_NCF_Sn[NCFSm].                                                                                                                          | LBIST                                                                             | No reaction                  |
| 38  | DFT2                                            | Test circuitry Group 2 activation. Fault is cleared by clearing the FCCU channel status, FCCU_NCF_Sn[NCFSm].                                                                                                                          | LBIST                                                                             | No reaction                  |
| 39  | DFT3                                            | Test circuitry Group 3 activation. Fault is cleared by clearing the FCCU channel status, FCCU_NCF_Sn[NCFSm].                                                                                                                          | LBIST                                                                             | No reaction                  |
| 40  | DFT4                                            | Test circuitry Group 4 activation. Fault is cleared by clearing the FCCU channel status, FCCU_NCF_Sn[NCFSm].                                                                                                                          | LBIST                                                                             | No reaction                  |
| 41  | Safety Core Machine Check Exception             | Safety Core exception indication. The fault is cleared by clearing the status in MCSR register of the Safety core followed by clearing the FCCU channel status, FCCU_NCF_Sn[NCFSm].                                                   | LBIST + willfully cause an exception (for example, access a non-existing address) | No reaction                  |

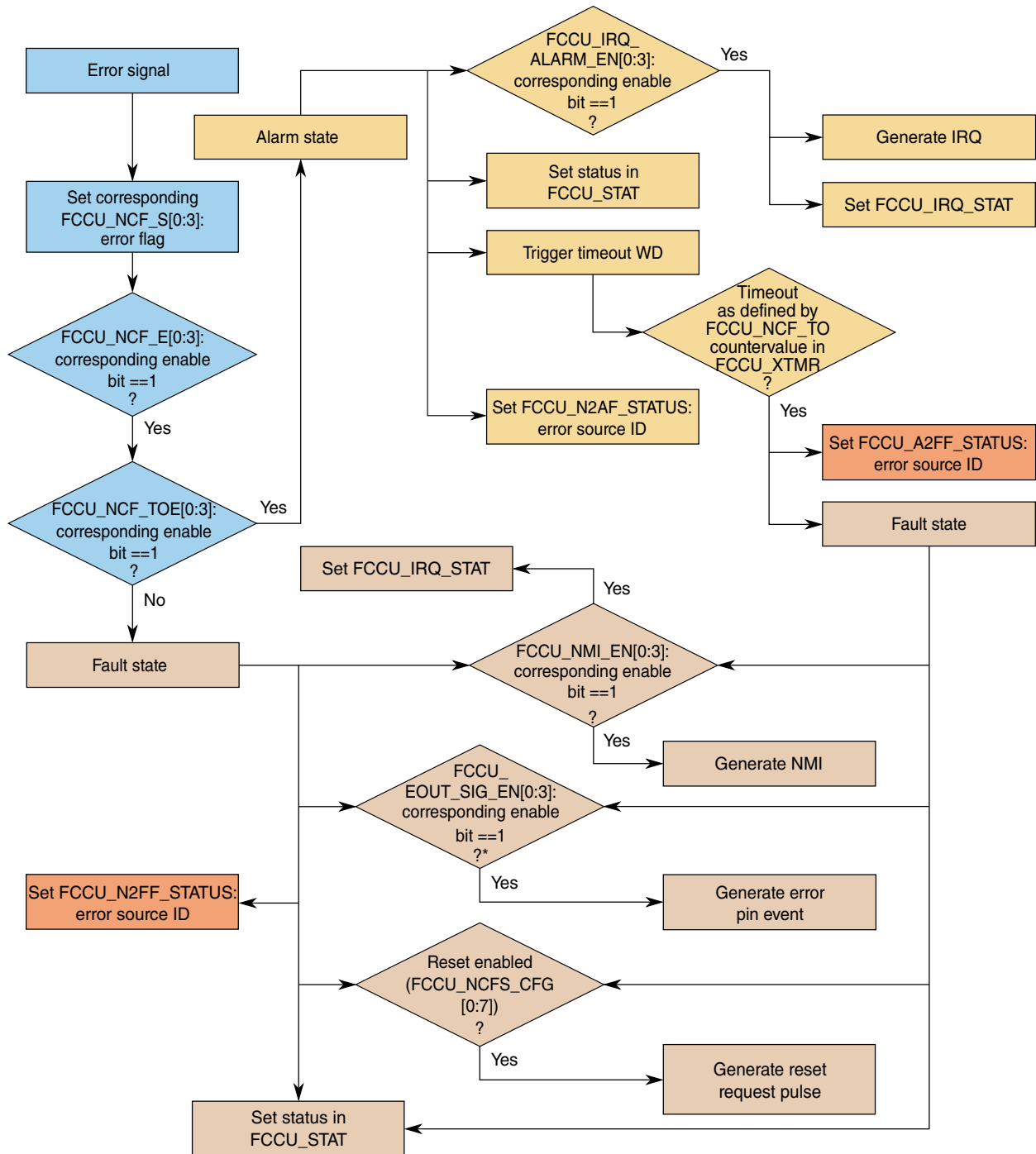
Table continues on the next page...

**Table 55-2. FCCU non-critical fault (NCF) mapping (continued)**

| NCF | Fault         | Fault description                                                                                                                                                                                | Error reaction path check <sup>1</sup>                                     | Default reaction after reset |
|-----|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------|------------------------------|
| 42  | Lockstep mode | Indication of disabled Checker Core or RCCUs. Fault is cleared by clearing the FCCU channel status, FCCU_NCF_Sn[NCFSm].                                                                          | LBIST                                                                      | No reaction                  |
| 43  | ERROR_IN_ERR  | Error input pin (external to the MCU) signaling an error condition. The fault is cleared by clearing the fault driving the pin followed by clearing the FCCU channel status, FCCU_NCF_Sn[NCFSm]. | External activation of EIN. External entity must pull the error input low. | No reaction                  |
| 44  | Unused        | —                                                                                                                                                                                                | —                                                                          | —                            |
| 45  | ERM_CE        | ERM module detected single ECC error. The fault is cleared by clearing the FCCU channel status, FCCU_NCF_Sn[NCFSm].                                                                              | EIM + LBIST + Feedback loop via FCCU Fake bit                              | No reaction                  |
| 46  | ERM_UCE       | ERM module detected multiple ECC error. The fault is cleared by clearing the FCCU channel status, FCCU_NCF_Sn[NCFSm].                                                                            | EIM + LBIST + Feedback loop via FCCU Fake bit                              | No reaction                  |

1. All channels can use feedback loop via FCCU Fake bit.
2. TDM does not have parity error during DCF loading. For UTEST Miscellaneous, only the DCF clients for PMC Digital have parity detection.
3. This fault occurs after any flash memory programming. To avoid unexpected behavior, configure the reaction to this fault carefully if flash memory programming is executed.
4. Fault 34 might be asserted during MBIST and LBIST execution. Before running MBISTs, configure this channel to result in no reset reaction. If this channel becomes asserted during MBIST testing, simply clear the fault and take no further action.

### 55.1.4 Error signal flow diagram



\* This condition must be satisfied only when FCCU is configured for Bistable fault-output mode (FCCU\_CFG[FOM]).

**Figure 55-1. Error signal flow**

### 55.1.5 Restriction on configuring RESET and NMI reactions on same FCCU channel

Do not configure an FCCU channel to have both a RESET reaction (using the FCCU\_NCFS\_CFG $n$  register) and an NMI reaction (using the FCCU\_NMI\_EN $n$  register) at the same time. If the FCCU channel is serviced and the chip resets, the NMI interrupt address is undefined and halts the chip.

### 55.1.6 FCCU\_CFG register event bit values by source (N and C)

For each bit in the [Configuration \(FCCU\\_CFG\)](#) register, this table specifies the values of  $N$  and  $C$  for this chip. For information on how to interpret and use these values, see [FCCU\\_CFG register bit value sources \(N and C\) by event](#).

| Bit | N | C |
|-----|---|---|
| 0   | 0 | 0 |
| 1   | 0 | 0 |
| 2   | 0 | 0 |
| 3   | 0 | 0 |
| 4   | 0 | 0 |
| 5   | 0 | 0 |
| 6   | 0 | 0 |
| 7   | — | 0 |
| 8   | — | 0 |
| 9   | — | 0 |
| 10  | — | 0 |
| 11  | — | 0 |
| 12  | — | 0 |
| 13  | — | 0 |
| 14  | — | 0 |
| 15  | — | 0 |
| 16  | 0 | 0 |
| 17  | 0 | 0 |
| 18  | 0 | 0 |
| 19  | — | 0 |
| 20  | 0 | 0 |
| 21  | 0 | 0 |
| 22  | 0 | 0 |
| 23  | 0 | 0 |
| 24  | 0 | 0 |

*Table continues on the next page...*

| Bit | N | C |
|-----|---|---|
| 25  | 0 | 0 |
| 26  | 0 | 0 |
| 27  | 0 | 0 |
| 28  | 0 | 1 |
| 29  | 0 | 0 |
| 30  | 0 | 0 |
| 31  | 0 | 0 |

## 55.2 Introduction

The Fault Collection and Control Unit (FCCU) offers a hardware channel to collect faults and to place the device into a safe state when a failure in the device is detected. No CPU intervention is requested for collection and control operation.

The FCCU offers a systematic approach to fault collection and control. The distinctive features of the module are:

- Redundant collection of hardware checker (for example, RCCU) results
- Redundant collection of fault information from safety relevant modules on the device
- Collection of test results
- Configurable fault control
- Configurable internal reactions for each NCF:
  - No reaction
  - IRQ
  - Functional reset
  - NMI
- External reactions via configurable output pins
- Watchdog timer for the reconfiguration phase
- Lockable configuration
- Nonvolatile memory interface (NVMCFG signals) for configuration loading
- One of the fault-output (EOUT) signals is high to indicate operational (OK) state (in bi-stable operation mode). The above is not true in case the error out protocol is configured to be a toggling protocol, for example, dual rail and time switching.

- After power on, the EOUT signals have high impedance.<sup>1</sup> They show operational state only after configuration by software.
- In case of a failure event or on software request for Error pin indication, the pin(s) are set to faulty state for a minimum time  $T_{min}$ , even if SW tries to release it before (for the case of error pin configured in Bi-stable mode only).
- The actual maximum time is 5 CLKSAFE cycles.

The FCCU circuitry is checked at the start-up by the self-checking procedure. The FCCU is operative with the default configuration immediately after the completion of the self-checking procedure. Internal (functional reset request pulse, interrupt request) and external (EOUT signaling) reactions are statically defined or programmable. The default configuration can be modified only in the CONFIG state. FCCU is designed to function when CLKSYS is faster than the CLKSAFE clocks.

## 55.2.1 Acronyms and abbreviations

| Term | Description                     |
|------|---------------------------------|
| CF   | Critical fault                  |
| CPU  | Central processing unit         |
| EOUT | Error out                       |
| FOSU | FCCU output supervision unit    |
| FSM  | Finite state machine            |
| INTC | Interrupt controller            |
| intf | Interface                       |
| IRQ  | Interrupt request               |
| NCF  | Noncritical fault               |
| NMI  | Nonmaskable interrupt           |
| NVM  | Nonvolatile memory              |
| RCC  | Redundancy control checkers     |
| RCCU | Redundancy control checker unit |
| SoC  | System-on-chip                  |
| STCU | Self-test control unit          |
| SW   | Software                        |

1. Actual value depends on device specific setting at pad level.

### 55.3 Main features

- Management of noncritical faults
- HW or SW fault recovery management
- Fault detection and collection
- Fault injection (fake faults)
- External reaction (fault state): EOUT signaling. Error indication via the pin(s) is controlled by the FCCU.
- Internal chip reactions (alarm state): interrupt request
- Internal chip reactions (fault state):
  - functional reset
  - NMI
- Bi-Stable, Dual-Rail and Time Switching output protocols on EOUT
- Internal (to the FCCU) watchdog timer for the reconfiguration phase
- Configuration lock
- Nonvolatile memory interface (NVMCFG signals) for configuration loading

### 55.4 Block diagram

The following figure is a top-level diagram of the FCCU module.

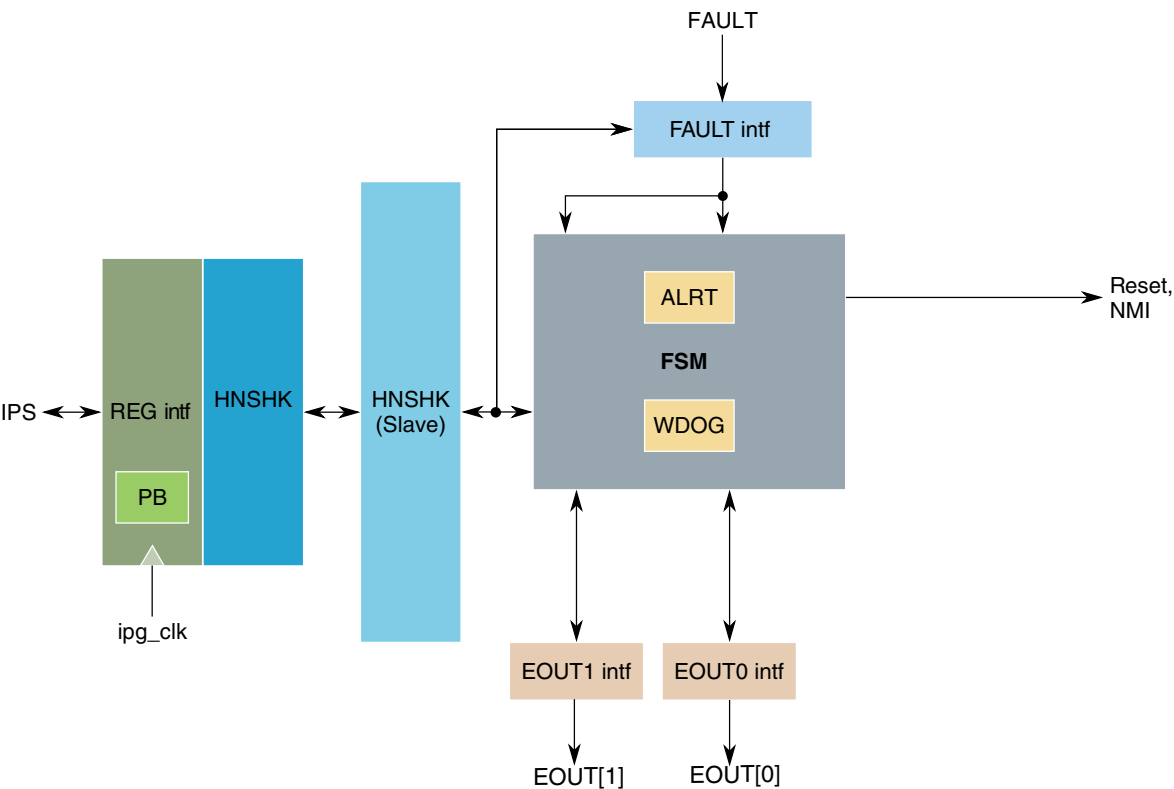


Figure 55-2. FCCU block diagram

This table describes the FCCU submodules.

Table 55-3. FCCU submodules

| Submodule                              | Description                                                                                                                                                                         |
|----------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| REG intf                               | Includes the register file, the IPS bus interface, the IRQ interface and the parity block (PB) for the configuration registers                                                      |
| HNSHK blocks (master and slave blocks) | Includes the FSM's ability to support the handshake between the REG interface and the FSM unit due to the usage of 2 asynchronous clocks (IPS system clock and RC oscillator clock) |
| FSM unit                               | Implements the main functions of the FCCU. The FSM also includes the: <ul style="list-style-type: none"><li>• Watchdog timer (WDG)</li><li>• Alarm timer (ALRT)</li></ul>           |
| FAULT intf                             | Implements the interface for the fault conditioning and management                                                                                                                  |
| EOUTx units                            | Implement the output stage to manage the EOUT interfaces                                                                                                                            |



## 55.5 Signal description

### 55.5.1 Signals

This table describes the signals on the boundary of FCCU.

| Signal                 | Direction | Function                                                                                                                                                                                                                                                                                                                                                                                 | Active level | Clock              |
|------------------------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|--------------------|
| <b>Clock and reset</b> |           |                                                                                                                                                                                                                                                                                                                                                                                          |              |                    |
| CLKSYS                 | Input     | System Clock—Supplies the primary clock.                                                                                                                                                                                                                                                                                                                                                 | Rising edge  | —                  |
| CLKSAFE0               | Input     | Safe Clock 0—Supplies one of two 16 MHz clocks for use by various redundant submodules. This clock and CLKSAFE1 are often referred to as the same clock (CLKSAFE).                                                                                                                                                                                                                       | Rising edge  | —                  |
| CLKSAFE1               | Input     | Safe Clock 1—Supplies one of two 16 MHz clocks for use by various redundant submodules. This clock and CLKSAFE0 are often referred to as the same clock (CLKSAFE).                                                                                                                                                                                                                       | Rising edge  | —                  |
| <b>EOUT interface</b>  |           |                                                                                                                                                                                                                                                                                                                                                                                          |              |                    |
| EIN[1:0]               | Input     | Error Input—Provide the mechanism for FCCU to capture the states that off-chip logic drives on the associated EOUT[1:0] signals. (On the chip, the EOUT signals are connected to their corresponding EIN signals.) FCCU captures the states in the <a href="#">EIN1</a> and <a href="#">EIN0</a> fields.                                                                                 | —            | Asynchronous       |
| EOUT[1:0]              | Output    | Error Output—Indicate FCCU's condition (Faulty, Nonfaulty, or Configuration) to off-chip logic.                                                                                                                                                                                                                                                                                          | Programmable | CLKSAFE0, CLKSAFE1 |
| <b>FOSU interface</b>  |           |                                                                                                                                                                                                                                                                                                                                                                                          |              |                    |
| FIF                    | Output    | <p>FCCU In Fault—Indicates to the FOSU module, if present, that FCCU is recognizing a fault. FCCU asserts and then deasserts this signal in two cases:</p> <p><i>Case 1</i></p> <p>When the EOUT signals are programmed to be always low (<a href="#">FCCU_SET_CLEAR</a>), FCCU asserts this signal until the EOUT signals are programmed not to be always low.</p> <p><i>Case 2</i></p> | High         | CLKSAFE0, CLKSAFE1 |

Table continues on the next page...

## Put FCCU in Configuration or Normal state

| Signal                     | Direction | Function                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | Active level | Clock  |
|----------------------------|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------|--------|
|                            |           | <p>When the EOUT signals are programmed not to be always low (<a href="#">FCCU_SET_CLEAR</a>), FCCU asserts this signal when all of the following are true:</p> <ul style="list-style-type: none"> <li>FCCU enters Fault state as the result of a fault.</li> <li>If the fault is a noncritical fault, and the EOUT signals are in a fault-output mode (<a href="#">FOM</a>) other than Bistable, the associated <a href="#">EOUTENx</a> field is set to enabled.</li> </ul> <p>FCCU deasserts this signal when FCCU leaves Fault state.</p> |              |        |
| <b>Interrupt interface</b> |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |              |        |
| NMIOUT                     | Output    | Nonmaskable Interrupt Output—Sends a nonmaskable interrupt request to the processor core or cores when FCCU enters Fault state as the result of a noncritical fault for which NMI is enabled as the reaction ( <a href="#">FCCU_NMI_ENa[NMIENx]</a> ).                                                                                                                                                                                                                                                                                       | Low          | CLKSYS |
| <b>NVM interface</b>       |           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |              |        |
| NVMCFG[21:0]               | Input     | Nonvolatile Memory Configuration—Provide the values of certain bits in the <a href="#">Configuration (FCCU_CFG)</a> register when certain events occur. See <a href="#">Configuration (FCCU_CFG)</a> . Some or all of these signals may be driven by bits in a data record in nonvolatile (flash) memory or may be tied high (1) or low (0). See <a href="#">FCCU_CFG register bit value sources (N and C) by event</a> .                                                                                                                    | Static value | —      |
| SSCMDONE                   | Input     | SSCM Done—After an FCCU reset, indicates that the SSCM module is now driving any NVMCFG signals that are associated with a data record stored in nonvolatile (flash) memory.                                                                                                                                                                                                                                                                                                                                                                 | High         | CLKSYS |

## 55.6 Put FCCU in Configuration or Normal state

## 55.6.1 Introduction

You put FCCU in Configuration state to change its configuration. You put FCCU in Normal state to save changes to the configuration.

After you configure FCCU, you can lock its configuration to prevent accidental changes.

## 55.6.2 About changing states

When changing states, keep the following in mind:

- To put FCCU in Configuration state, FCCU must be in Normal state.
- After FCCU is reset, the configuration is temporarily locked. You must temporarily unlock the configuration before you can put FCCU in Configuration state.
- When FCCU is in Configuration state, FCCU doesn't actually save the changes you make to the configuration. To save changes to the configuration, you must manually put FCCU in Normal state. If FCCU automatically leaves Configuration state and enters Normal state because the configuration-timer interval (**TO**) expires (called a Configuration-state timeout), FCCU changes the value of the [Configuration \(FCCU\\_CFG\) register](#) to its Configuration-state-timeout value and the value of each of the other configuration registers to its reset value. For information on the Configuration-state timeout value, see [FCCU\\_CFG register bit value sources \(N and C\) by event](#). For a list of configuration registers, see [Configuration registers](#).

## 55.6.3 About locking the configuration

When locking the configuration, keep the following in mind:

- If you attempt to lock the configuration while FCCU is in Configuration state, FCCU doesn't actually lock the configuration until FCCU leaves Configuration state—that is, either you put FCCU in Normal state, or FCCU puts FCCU in Normal state because the configuration-timer interval (**TO**) expires.
- After you permanently lock the configuration, you must reset FCCU before you can put FCCU in Configuration state.

## 55.6.4 Put FCCU in Configuration state

1. From Supervisor mode, temporarily unlock the configuration ([TRANSKEY](#)).
2. To determine FCCU's state, run the OP3 operation (see [Run an operation](#)).
3. Check the operation status ([OPS](#)).
  - If the operation status is Successful, go to step 4.
  - If the operation status is Aborted, go to step 2.

4. Check the FCCU status ([STATUS](#)).
  - If the FCCU status is Configuration, FCCU is in Configuration state. Stop.
  - If the FCCU status is Normal, go to step [5](#).
  - If the FCCU status is Alarm or Fault, go to step [7](#).
5. To put FCCU in Configuration state, run the OP1 operation (see [Run an operation](#)).
6. Check the operation status ([OPS](#)).
  - If the operation status is Successful, FCCU is in Configuration state. Stop.
  - If the operation status is Aborted, the configuration is probably permanently locked. Go to step [9](#).
7. Recover all faults.
8. Go to step [2](#).
9. Reset FCCU.

### 55.6.5 Put FCCU in Normal state

1. To determine FCCU's state, run the OP3 operation (see [Run an operation](#)).
2. Check the operation status ([OPS](#)).
  - If the operation status is Successful, go to step [3](#).
  - If the operation status is Aborted, go to step [1](#).
3. Check the FCCU status ([STATUS](#)).
  - If the FCCU status is Normal, FCCU is in Normal state. Go to step [8](#).
  - If the FCCU status is Configuration, go to step [4](#).
  - If the FCCU status is Alarm or Fault, go to step [6](#).
4. To put FCCU in Normal state, run the OP2 operation (see [Run an operation](#)).
5. Check the operation status ([OPS](#)).
  - If the operation status is Successful, FCCU is in Normal state. Go to step [8](#).
  - If the operation status is Aborted, go to step [4](#).
6. Recover all faults.
7. Go to step [1](#).
8. If you want to prevent changes to the configuration, lock it:
  - To require a key to unlock the configuration, from Supervisor mode, temporarily lock the configuration ([TRANSKEY](#)).
  - To require a reset of FCCU to unlock the configuration, from Supervisor mode, permanently lock the configuration ([PERMNTKEY](#)).

The configuration is permanently locked until FCCU is reset.

## 55.7 Run operations

### 55.7.1 Introduction

You run operations to perform actions such as putting FCCU in Configuration state or reading a timer. For a complete list of operations you can run, see [OPR](#).

### 55.7.2 About running operations

When running operations, keep the following in mind:

- FCCU ignores any operations initiated while the operation status is In Progress.
- Certain operations must be unlocked before you can initiate them. After you initiate them, they are again locked.

### 55.7.3 Run an operation

1. Check the operation status ([OPS](#)).
2. If the operation status is In Progress, go to step 1.
3. If the operation must be unlocked before you can initiate it, unlock the operation ([CTRLK](#)) using a 32-bit write.

#### NOTE

The [Control Key \(FCCU\\_CTRLK\)](#) register used in this step and the [Control \(FCCU\\_CTRL\)](#) register used in the next step must be written with consecutive instructions. Do not use read-modify-write instructions, such as bit-field instructions, to modify these registers.

4. Initiate the operation ([OPR](#)) using a 32-bit write.
5. Check the operation status ([OPS](#)).
6. If the operation status is In Progress, go to step 5.

The operation status is now Idle (OP15 only), Aborted, or Successful.

## 55.8 Functional description

### 55.8.1 Definitions

In general, the following definitions are applicable for the fault management:

- **HW recoverable fault:** the fault indication is an edge-triggered and level-sensitive signal that remains asserted until the fault cause is deasserted. That is, if logical 0 on the fault signal indicates fault, then the status flags are valid as long as the fault line stays at 0. The status is automatically cleared when the fault signal goes to 1. Typically the fault signal is latched external to the FCCU in the module where the fault occurred. The FCCU state transitions are consequently executed on the state changes of the input fault signal. No SW intervention in the FCCU is required to recover the fault condition.
- **SW recoverable fault:** the fault indication is a signal asserted without a defined time duration. The fault signal is latched in the FCCU. The fault recovery is executed following a SW recovery procedure (status/flag register clearing).

HW recoverable is an option to exclude the handling of error source/s by FCCU management SW, in case it is known that the fault is recoverable by itself when the fault condition gets corrected.

### 55.8.2 FSM description

FCCU functionality is depicted by the finite state machine (FSM) state diagram.

Basically four states are identified with the following meaning:

- **Configuration:** Used only to modify the configuration of FCCU from its default. A subset of the FCCU registers, dedicated to define the FCCU configuration (global configuration, reactions to fault, timeout, noncritical fault masking) can be accessed in write mode only in Configuration state.

Configuration state is accessible only in Normal state and if the configuration is not locked. The permanent configuration lock can be disabled by a reset which also resets the FCCU., The transient lock register is unlocked by writing BCh into it. FCCU gets transiently locked again if an invalid key is written into FCCU\_TRANS\_LOCK register (that is, other than BCh). Key to lock FCCU permanently is FFh, written in the FCCU\_PERMNT\_LOCK register.

After the release of reset the state of the transient lock will be locked. The state of the permanent lock will be unlocked.

The locking feature only restricts the FSM movement into Configuration state. Once the user enters Configuration state and then tries to lock the configuration, the locking of configuration will be effective after FCCU moves to Normal state, it will not be effective in the current Configuration state.

The Configuration to Normal state transition can be executed by SW or automatically following a timeout condition of the watchdog. In case the timeout information and the SW request to mode change to Normal appears at the same time, watchdog timeout has the priority and hence the Configuration registers (those that are writable only in Configuration state) are reset to their default values. The movement to Normal is made.

The incoming faults, occurring during the configuration phase (Configuration state) are latched in order to process them when FCCU is moved into Normal state, according to the new configuration.

All pending faults that occur during Configuration state result in both of the following:

- Highest-priority state transition
- Interrupt generation (NMI or alarm IRQ)

If the state transition occurs, it gives the reset reaction corresponding to the worst case based on all the faults (pending or nonpending faults) that occurred during Configuration state.

- Normal: This is FCCU's operating state when no faults are occurring. It is also the default state on the reset exit. Following state transitions occur on one of the following events:
  - unmasked noncritical faults with the timeout disabled → FCCU moves to Fault state
  - unmasked noncritical faults with the timeout enabled → FCCU moves to Alarm state
  - masked noncritical faults → FCCU stays in Normal state
- Alarm: FCCU moves into Alarm state when an unmasked noncritical fault occurs and the timeout is enabled. The transition to Alarm state goes along with an interrupt alarm, if enabled. By definition, this fault may be recovered within a programmable

timeout period, before it generates a transition to Fault state. The timeout is reinitialized if FCCU enters Normal state. The timeout restarts following the recovery from Fault state.

- Fault: FCCU moves into Fault state when one of the following condition occurs:
  - timeout related to a noncritical fault when FCCU is in Alarm state
  - unmasked noncritical faults with the timeout disabled

The transition from Normal/Alarm to Fault state goes along with the generation of:

- NMI interrupt (optional)
- EOUT signaling (optional)
- SW option: functional reset
- Non Maskable Interrupt (NMI) is routed only to the Safety Core

After moving to Fault state, if there is either a previous pending fault or a new fault for which NMI is enabled, NMI generation takes place.

Multiple faults can occur at the same time. Consider the case where there is a fault for which Alarm state is enabled and IRQ is programmed along with a fault for which Alarm state is disabled. In that case, the FSM moves to Fault state, but with the generation of alarm IRQ for the fault which has IRQ enabled.



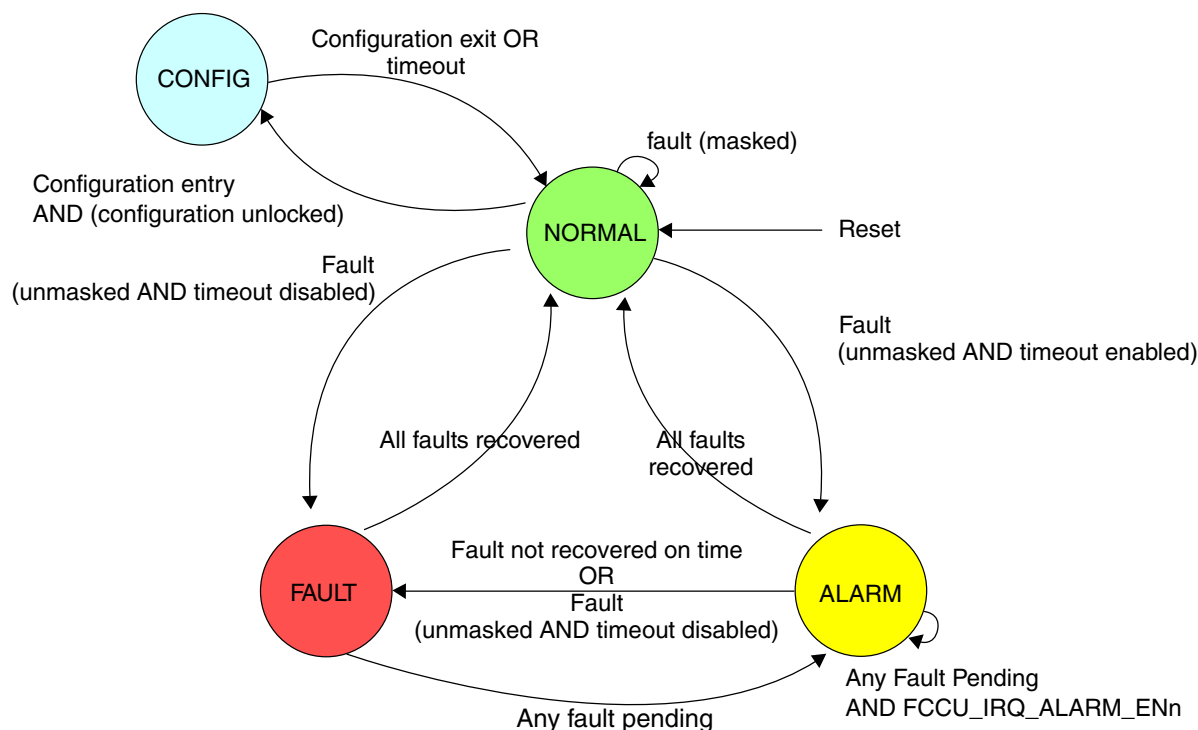


Figure 55-3. FCCU state diagram

### 55.8.3 Fault priority scheme and nesting

The FAULT state has a higher priority than the ALARM state in case of concurrent fault events (noncritical) that occur in the NORMAL state.

The ALARM to FAULT state transition occurs if a noncritical fault (unmasked and with time-out disabled) is asserted in the ALARM state.

The ALARM to NORMAL state transition occurs only if all the noncritical faults (including the faults that have been collected after the entry in the ALARM state) have been cleared (SW or HW recovery); otherwise the FCCU will remain in the ALARM state.

The FAULT to NORMAL state transition occurs only if all the noncritical faults (including the faults that have been collected after the entry in the FAULT/ALARM state) have been cleared (SW or HW recovery); otherwise the FCCU will move to the ALARM state (if any noncritical fault is still pending and the time-out is not elapsed).

In general, no fault nesting is supported except for the noncritical faults that cause an ALARM to FAULT state transition. In this case the NCF timer is stopped until the FAULT state is recovered. If FCCU is in ALARM state and another fault comes, which has its alarm timeout enabled, then the alarm timer shall not reload and shall not start again.

## 55.8.4 STCU interface

The STCU interface includes:

- A set of signals resulting from the self-checking procedure connected externally at the FCCU.

The STCU fault signals are processed by the FCCU when the SoC is re-booted following the self-testing procedure. The STCU includes a status register that stores the self-testing results (flags).

- a mask that inhibits the EOUT dummy signaling until the STCU self-checking procedure has been completed.

During the self-testing procedure, depending on the STCU results, three cases are applicable:

1. STCU completes the self-testing procedure successfully. The SoC reboots and the FCCU is responsible for fault collection and indication.
2. STCU completes the self-testing procedure with low severity failures. The FCCU records the fault condition but does not, by default, trigger a reset cycle.
3. STCU completes the self-testing procedure with serious failures. The FCCU triggers a reset cycle in response to this fault input from the STCU. Because the fault condition survives reset, new reset cycles are subsequently triggered until the reset escalation feature takes control, freezing the chip in the reset state until a POR.

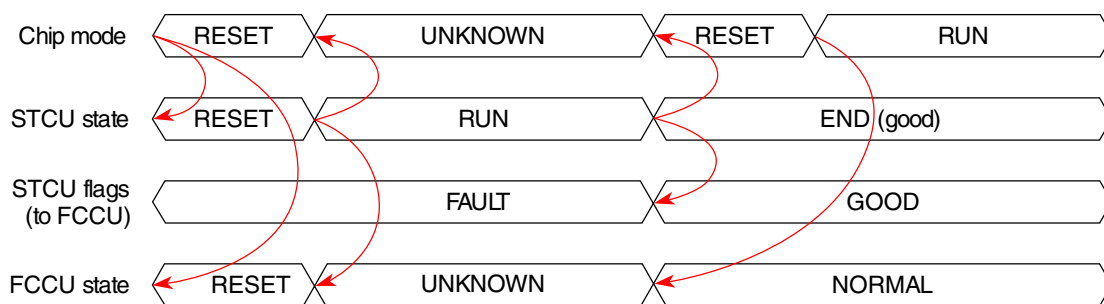
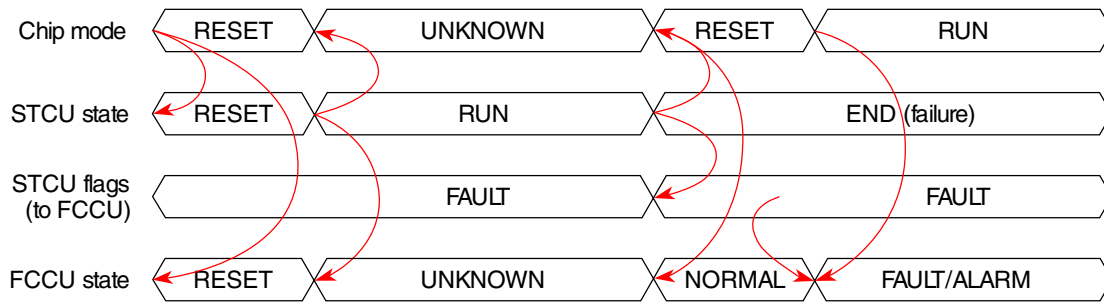
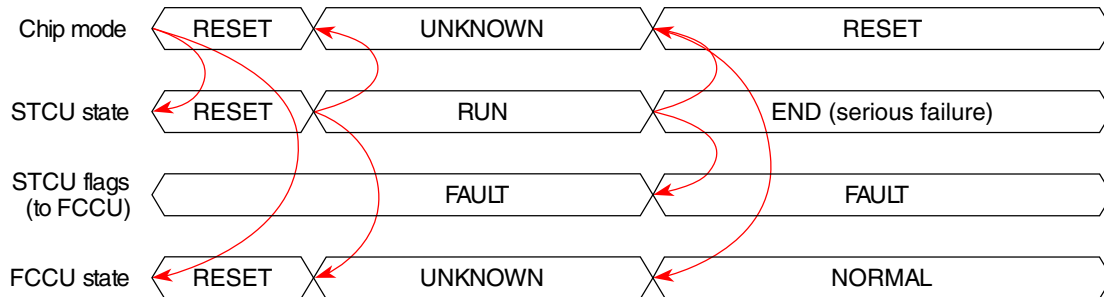


**Figure 55-4. STCU-FCCU (case 1)**

**Figure 55-5. STCU-FCCU (case 2)****Figure 55-6. STCU-FCCU (case 3)**

### 55.8.5 Nonvolatile memory interface

FCCU includes a nonvolatile memory interface (NVMCFG signals) that provides the values of certain bits in the [Configuration \(FCCU\\_CFG\)](#) register when certain events occur. See [Configuration \(FCCU\\_CFG\)](#). Whether any of these signals are actually connected to nonvolatile (flash) memory is chip-specific. Some or all of these signals may be driven by bits in a data record in nonvolatile (flash) memory or may be tied high (1) or low (0). See [FCCU\\_CFG register bit value sources \(N and C\) by event](#).

The timing diagram related to the NVM interface is given in [Figure 55-7](#).

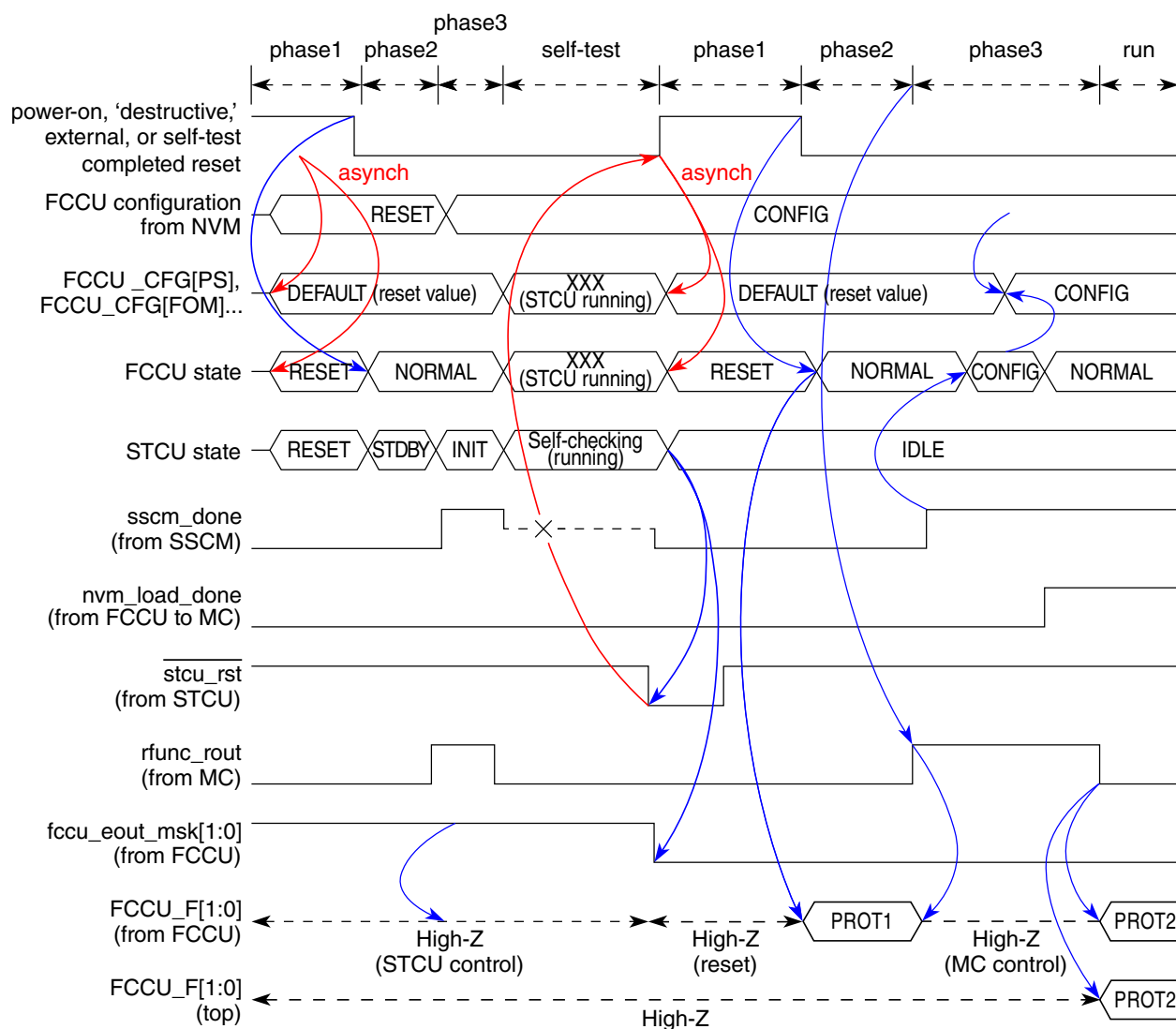


Figure 55-7. NVM interface

## 55.8.6 EOUT interface

### 55.8.6.1 Introduction

You use the EOUT[1:0] signals to indicate FCCU's condition to off-chip logic. For information on the availability and names of these FCCU signals on the boundary of this chip, see the chip-specific FCCU information.

### 55.8.6.2 The FCCU conditions

There are three FCCU conditions:

| Condition     | Description                                                                                                                                                                   |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Faulty        | The fault-output (EOUT) timer is running (see <a href="#">How the fault-output (EOUT) timer works</a> ).                                                                      |
| Nonfaulty     | All of the following are true: <ul style="list-style-type: none"> <li>• The fault-output (EOUT) timer is not running.</li> <li>• FCCU is in Alarm or Normal state.</li> </ul> |
| Configuration | All of the following are true: <ul style="list-style-type: none"> <li>• The fault-output (EOUT) timer is not running.</li> <li>• FCCU is in Configuration state.</li> </ul>   |

### 55.8.6.3 How the fault-output (EOUT) timer works

FCCU starts the fault-output (EOUT) timer when all of the following are true:

- The fault-output (EOUT) signals are not programmed to be always low ([FCCU\\_SET\\_CLEAR](#)).
- The fault-output (EOUT) timer isn't already running.
- FCCU enters Fault state as the result of a fault.

When the fault-output (EOUT) timer is already running and a new fault occurs:

- When FCCU is in Configuration state: FCCU doesn't restart the fault-output (EOUT) timer.
- When FCCU is in Normal state or Alarm state:
  - And Alarm state is enabled for the fault (noncritical): FCCU enters (or remains in) Alarm state but doesn't restart the fault-output (EOUT) timer.
  - And Alarm state is disabled for the fault (noncritical): FCCU enters Fault state and restarts the fault-output (EOUT) timer.
- When FCCU is in Fault state: FCCU restarts the fault-output (EOUT) timer.

FCCU stops and reinitializes the fault-output (EOUT) timer when all of the following are true:

- If the EOUT signals are in Bistable fault-output mode ([FOM](#)), T\_min (see [DELTA\\_T](#)) has expired.
- All faults that caused FCCU to enter or remain in Fault state since FCCU started the fault-output (EOUT) timer have been cleared, causing FCCU to return to Normal state.

#### 55.8.6.4 Prepare the EOUT signals to indicate FCCU's condition

- If the EOUT signals are in Bistable fault-output mode ([FOM](#)), make sure the EOUT signals are controlled by FCCU's finite state machine ([FCCU\\_SET\\_CLEAR](#)).
- Make sure the EOUT signals are active ([FCCU\\_SET\\_AFTER\\_RESET](#)).

#### NOTE

If the EOUT signals are in Bistable fault-output mode, you must deactivate and then reactivate the EOUT signals ([FCCU\\_SET\\_AFTER\\_RESET](#)) to correctly initialize them so they have opposite states.

#### 55.8.6.5 More about the EOUT interface

Different fault-output modes (protocols) for the fault-output (EOUT) interface are supported ([FCCU\\_CFG\[FOM\]](#)):

- Dual-Rail
- Time-Switching
- Bistable
- Test

You can further configure the fault-output modes using the following attributes:

| Attribute          | Field                        | Setting used in the example diagrams and tables that follow |
|--------------------|------------------------------|-------------------------------------------------------------|
| Switching mode     | <a href="#">FCCU_CFG[SM]</a> | Slow                                                        |
| Polarity selection | <a href="#">FCCU_CFG[PS]</a> | For the faulty indication, EOUT1 is high, and EOUT0 is low. |

The EOUT frequency is generated by dividing the CLKSAFE frequency by a fixed factor of  $2^{18}$ .

With a CLKSAFE frequency of 16 MHz, this drives a signal of 61 Hz on EOUT.

The external monitor of the EOUT protocol should oversample the EOUT signals in order to synchronize periodically the external clock (used by the monitor) and CLKSAFE (which detects the edge transition of the EOUT protocol) in Dual-Rail or Time-Switching fault-output mode.

In case of a failure event or on software request for EOUT indication, the signal(s) are set to the faulty state for a minimum time  $T_{min}$ , even if software tries to release it before. If software configures the error pins to OK(1), and if a fault comes trying to drive the pin to NOK(0), then priority is given to the fault indication and the error signals indicate NOK, such as an incoming fault is not masked when software has set the error signal to high. During the  $T_{min}$  by a non-software fault, the FCCU FSM moves independently of this signal state (low), and as soon as the timer expires, the pin behavior is dictated by the state in which the FSM finds itself in, and it is not possible to set the signals to OK by software moving FCCU to Configuration state, as long as this timer is running. No software intervention is needed to bring the signal from the low state.

Software can bring the pin back to OK state by clearing the faults and waiting for the  $T_{min}$  interval to expire, after which the FCCU automatically enters Normal state and the error signal indicates OK.

In case another failure event happens within  $T_{min}$  after a first one, the  $T_{min}$  counter is restarted.

### NOTE

The transition from Fault to Configuration state is not possible but is shown to display the behavior in all four states—reset, normal, error, and configuration.

### NOTE

FCCU remains in Normal state when a fault is disabled; therefore, NA (not applicable) appears in the following table.

### NOTE

The following table is valid only when the EOUT signals are active ([FCCU\\_SET\\_AFTER\\_RESET](#)).

**Table 55-4. FCCU error pin behavior on state transitions**

| Fault signaling state | Transition from Normal to Fault state |                  |           | Normal+Configuration states |                  |           | Transition from Normal to Alarm state |                  |           |
|-----------------------|---------------------------------------|------------------|-----------|-----------------------------|------------------|-----------|---------------------------------------|------------------|-----------|
|                       | with Fault disabled   enabled         |                  |           |                             |                  |           | with Fault disabled   enabled         |                  |           |
|                       | Internal error pin                    | Error pin enable | Error pad | Internal error pin          | Error pin enable | Error pad | Internal error pin                    | Error pin enable | Error pad |
| Disabled              | NA   OK                               | NA   1           | NA   OK   | OK                          | 1                | OK        | NA   OK                               | NA   1           | NA   OK   |
| Enabled               | NA   NOK                              | NA   1           | NA   NOK  | OK                          | 1                | OK        | NA   OK                               | NA   1           | NA   OK   |

**NOTE**

The first time FCCU is in Configuration state, the EOUT pads are in a high-impedance state; thereafter, their states reflect the programmed EOUT protocol.

**55.8.6.6 Dual-rail protocol**

Dual-rail encoding is an alternate method for encoding bits. In contrast with classical encoding, where each signal carries a single-bit value, dual-rail encoded circuits use two wires to carry each bit. The encoding scheme is given in [Table 55-5](#) and the related timing diagram is given in [Figure 55-8](#) and [Figure 55-9](#).

**Table 55-5. Dual-rail encoding**

| Logical state | Dual-rail encoding (output pins EOUT[1:0]) | Note        |
|---------------|--------------------------------------------|-------------|
| non-faulty    | 10                                         | toggling    |
| non-faulty    | 01                                         |             |
| faulty        | 00                                         | toggling    |
| faulty        | 11                                         |             |
| reset         | Hi impedance <sup>1</sup>                  | no toggling |
| configuration | same as NORMAL                             | toggling    |

1. Final value depends on the SoC setting at pad level.

As long as FCCU is in NORMAL or ALARM state, output will show "non-faulty" signal. Output pins EOUT[0:1] will toggle between 01 and 10 with a given frequency of 61 Hz.

In the RESET phase the output pins are set as "high impedance".

**Note**

[Figure 55-8](#) and [Figure 55-9](#) are formatted to display the behavior in all four phases (reset, normal, error, and config), not to imply transitions between one phase to another. In particular, transition from error phase to config phase is not possible.



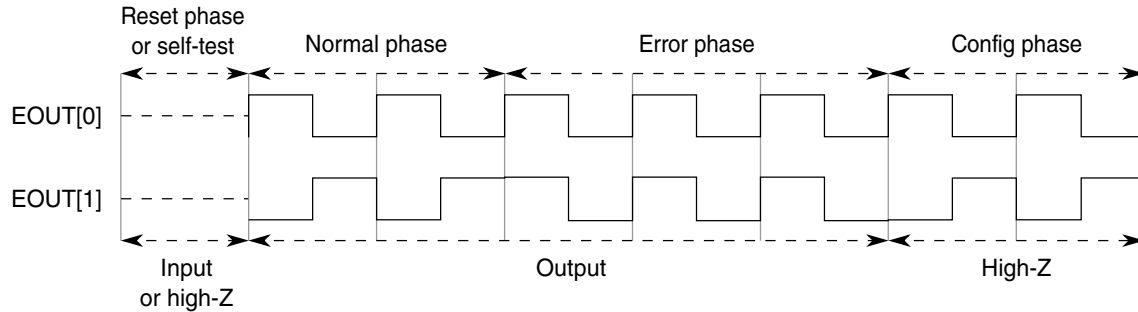


Figure 55-8. Dual-rail protocol (slow switching mode)

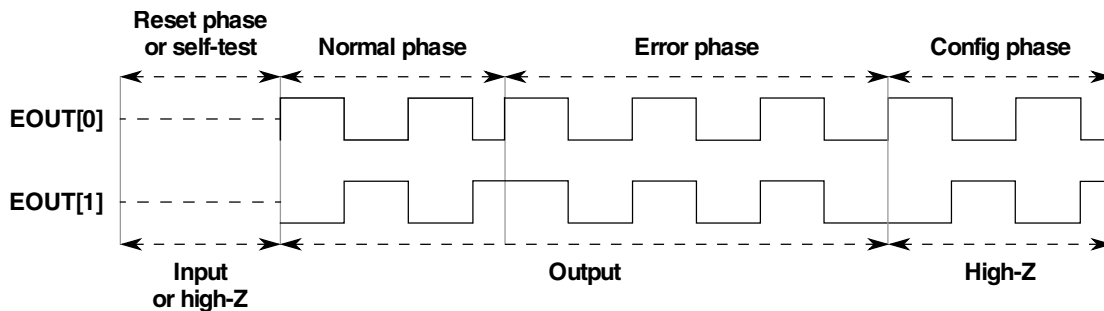


Figure 55-9. Dual-rail protocol (fast switching mode)

### 55.8.6.7 Time-switching protocol

The encoding scheme is given in [Table 55-6](#) and the related timing diagram is given in [Figure 55-10](#).

Table 55-6. Time-switching encoding

| Logical state | Time-switching encoding (output pins EOUT[1:0]) | Note        |
|---------------|-------------------------------------------------|-------------|
| non-faulty    | 10                                              | toggling    |
| non-faulty    | 01                                              |             |
| faulty        | 10                                              | no toggling |
| reset         | high-Z <sup>1</sup>                             | no toggling |
| configuration | same as NORMAL                                  | toggling    |

1. Final value depends on device specific settings at pad level.

As long as FCCU is in NORMAL or ALARM state, outputs will show "non-faulty" signal. Output pins #0, #1 will toggle between 01 and 10 with frequency of 61 Hz.

In the FAULT state, the output pin EOUT[0] is set as low.

In Time Switching mode the second output (EOUT[1]) is the inverted signal of first output (EOUT[0]).

In the RESET phase the output pins are set as "high impedance".<sup>2</sup>

**Note**

Figure 55-10 is formatted to display the behavior in all four phases (reset, normal, error, and config), not to imply transitions between one phase to another. In particular, transition from error phase to config phase is not possible.

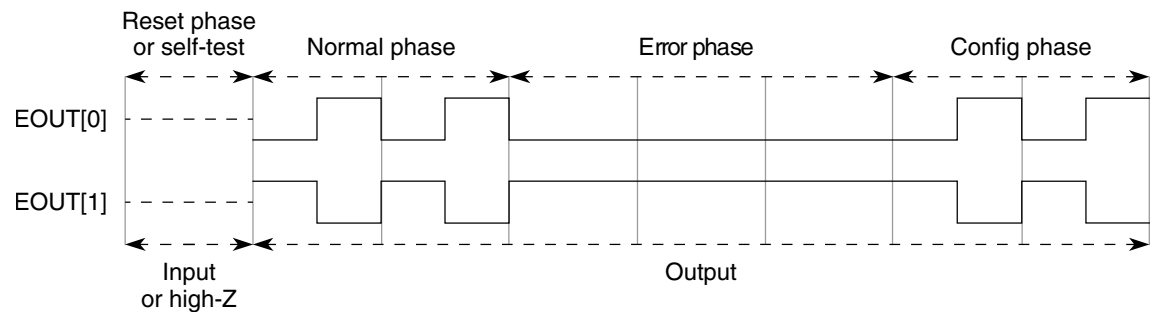


**Figure 55-10. Time-switching protocol**

**55.8.6.8 Bi-stable protocol**

The encoding scheme is given in Table 55-7 and the related timing diagram is given in Figure 55-11.

**Table 55-7. Bi-stable encoding**

| Logical state | Bi-stable encoding (output pins EOUT[1:0]) | Note        |
|---------------|--------------------------------------------|-------------|
| non-faulty    | 01                                         | no toggling |
| faulty        | 10                                         | no toggling |
| reset         | high-Z <sup>1</sup>                        | no toggling |
| configuration | same as NORMAL                             | no toggling |

1. Final value depends on device specific settings at pad level.

In the FAULT state, the faulty logical state is indicated. In NORMAL or ALARM state, "no-faulty" state is indicated. In Bi-stable mode the second output (EOUT[1]) is the inverted signal of first output (EOUT[0]).

In the RESET phase the output pins are set as "high impedance".<sup>3</sup>

2. Actual value depends on device specific settings at pad level.

3. Actual value depends on device specific settings at pad level.

## Note

Figure 55-11 is formatted to display the behavior in all four phases (reset, normal, error, and config), not to imply transitions between one phase to another. In particular, transition from error phase to config phase is not possible.

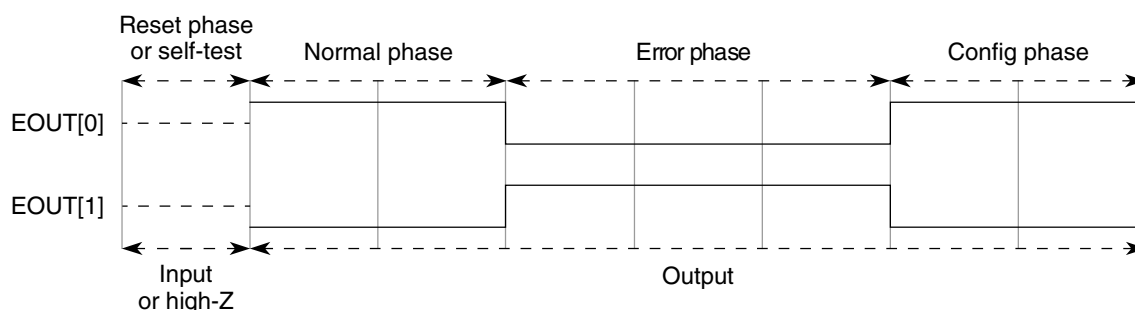


**Figure 55-11. Bi-stable protocol**

### 55.8.7 Error signal flow

See the chip-specific FCCU information for the diagram of the signal flow.

## 55.9 Register descriptions

The FCCU registers are listed in the table below. Any address offset not explicitly mentioned in this table is reserved.

The FCCU supports word (32-bit), half-word (16-bit), and byte (8-bit) read and write accesses.

Follow these register-access guidelines:

- Don't read from or write to any addresses that are not shown in the following table. Doing so may result in a transfer error.
- Don't write to any of the configuration registers unless FCCU is in Configuration state. Doing so results in a transfer error.
- Don't write to FCCU\_TRANS\_LOCK or FCCU\_PERMNT\_LOCK unless your code runs in Supervisor mode. Doing so results in a transfer error.

## Register descriptions

All the registers accessible in write mode only in CONFIG state are referred as configuration registers. These configuration registers return to the default value after configuration watchdog timer expires. These are the registers protected by FCCU\_TRANS\_LOCK and FCCU\_PERMNT\_LOCK registers. The configuration register setting has effect only when the FCCU state exits from the CONFIG state.

For each possible NCF failure source, a different reaction—including no reaction—is configurable through the use of NMI, IRQ, and reset selection registers. It is not possible for a single event upset to switch off all reactions on failures as implementation is per fault source (but it will be possible to switch them all off by SW if intended). Failures themselves are not able to disable all reactions and indications.

The FCCU is not reset by functional resets.

### FCCU memory map

| Address offset (hex) | Register name                                          | Width (in bits) | Access | Reset value                 | Section/ page                |
|----------------------|--------------------------------------------------------|-----------------|--------|-----------------------------|------------------------------|
| 0                    | Control (FCCU_CTRL)                                    | 32              | R/W    | 0000_00C0h                  | <a href="#">55.9.1/2522</a>  |
| 4                    | Control Key (FCCU_CTRLK)                               | 32              | W      | 0000_0000h                  | <a href="#">55.9.2/2524</a>  |
| 8                    | Configuration (FCCU_CFG)                               | 32              | R/W    | <a href="#">See section</a> | <a href="#">55.9.3/2525</a>  |
| 1C                   | Noncritical Fault Configuration (FCCU_NCF_CFG0)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">55.9.4/2528</a>  |
| 20                   | Noncritical Fault Configuration (FCCU_NCF_CFG1)        | 32              | R/W    | <a href="#">See section</a> | <a href="#">55.9.4/2528</a>  |
| 4C                   | Noncritical Fault State Configuration (FCCU_NCFS_CFG0) | 32              | R/W    | <a href="#">See section</a> | <a href="#">55.9.5/2529</a>  |
| 50                   | Noncritical Fault State Configuration (FCCU_NCFS_CFG1) | 32              | R/W    | <a href="#">See section</a> | <a href="#">55.9.5/2529</a>  |
| 54                   | Noncritical Fault State Configuration (FCCU_NCFS_CFG2) | 32              | R/W    | <a href="#">See section</a> | <a href="#">55.9.5/2529</a>  |
| 80                   | Noncritical Fault Status (FCCU_NCF_S0)                 | 32              | R/W    | 0000_0000h                  | <a href="#">55.9.6/2530</a>  |
| 84                   | Noncritical Fault Status (FCCU_NCF_S1)                 | 32              | R/W    | 0000_0000h                  | <a href="#">55.9.6/2530</a>  |
| 90                   | Noncritical Fault Key (FCCU_NCFK)                      | 32              | W      | 0000_0000h                  | <a href="#">55.9.7/2532</a>  |
| 94                   | Noncritical Fault Enable (FCCU_NCF_E0)                 | 32              | R/W    | <a href="#">See section</a> | <a href="#">55.9.8/2533</a>  |
| 98                   | Noncritical Fault Enable (FCCU_NCF_E1)                 | 32              | R/W    | <a href="#">See section</a> | <a href="#">55.9.8/2533</a>  |
| A4                   | Noncritical Fault Timeout Enable (FCCU_NCF_TOE0)       | 32              | R/W    | <a href="#">See section</a> | <a href="#">55.9.9/2534</a>  |
| A8                   | Noncritical Fault Timeout Enable (FCCU_NCF_TOE1)       | 32              | R/W    | <a href="#">See section</a> | <a href="#">55.9.9/2534</a>  |
| B4                   | Noncritical Fault Timeout (FCCU_NCF_TO)                | 32              | R/W    | <a href="#">See section</a> | <a href="#">55.9.10/2535</a> |
| B8                   | Configuration-State Timer Interval (FCCU_CFG_TO)       | 32              | R/W    | <a href="#">See section</a> | <a href="#">55.9.11/2535</a> |
| BC                   | IO Control (FCCU_EINOUT)                               | 32              | R/W    | <a href="#">See section</a> | <a href="#">55.9.12/2536</a> |
| C0                   | Status (FCCU_STAT)                                     | 32              | R      | 0000_0010h                  | <a href="#">55.9.13/2538</a> |
| C4                   | NA Freeze Status (FCCU_N2AF_STATUS)                    | 32              | R      | 0000_0000h                  | <a href="#">55.9.14/2540</a> |
| C8                   | AF Freeze Status (FCCU_A2FF_STATUS)                    | 32              | R      | 0000_0000h                  | <a href="#">55.9.15/2541</a> |

Table continues on the next page...

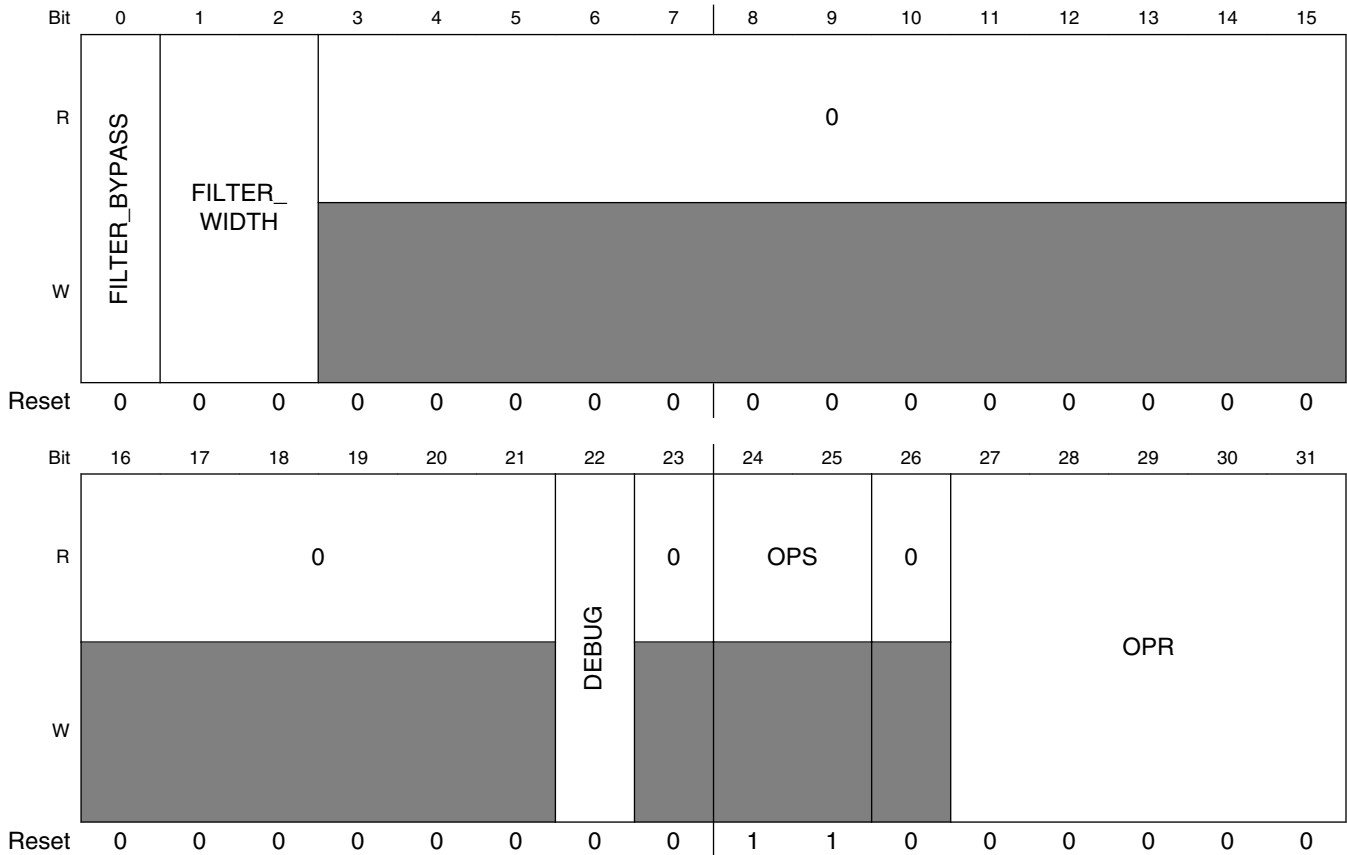
## FCCU memory map (continued)

| Address offset (hex) | Register name                                                     | Width (in bits) | Access | Reset value | Section/ page                 |
|----------------------|-------------------------------------------------------------------|-----------------|--------|-------------|-------------------------------|
| CC                   | NF Freeze Status (FCCU_N2FF_STATUS)                               | 32              | R      | 0000_0000h  | <a href="#">55.9.16/ 2542</a> |
| D0                   | FA Freeze Status (FCCU_F2A_STATUS)                                | 32              | R      | 0000_0000h  | <a href="#">55.9.17/ 2543</a> |
| DC                   | Noncritical Fault Fake (FCCU_NCFF)                                | 32              | R/W    | 0000_0000h  | <a href="#">55.9.18/ 2544</a> |
| E0                   | IRQ Status (FCCU_IRQ_STAT)                                        | 32              | R/W    | 0000_0000h  | <a href="#">55.9.19/ 2545</a> |
| E4                   | IRQ Enable (FCCU_IRQ_EN)                                          | 32              | R/W    | 0000_0000h  | <a href="#">55.9.20/ 2546</a> |
| E8                   | X Timer (FCCU_XTMR)                                               | 32              | R      | 0000_0000h  | <a href="#">55.9.21/ 2547</a> |
| F0                   | Transient Configuration Lock (FCCU_TRANS_LOCK)                    | 32              | W      | 0000_0000h  | <a href="#">55.9.22/ 2548</a> |
| F4                   | Permanent Configuration Lock (FCCU_PERMNT_LOCK)                   | 32              | W      | 0000_0000h  | <a href="#">55.9.23/ 2549</a> |
| F8                   | Delta T (FCCU_DELTA_T)                                            | 32              | R/W    | 0000_0000h  | <a href="#">55.9.24/ 2549</a> |
| FC                   | IRQ Alarm Enable (FCCU_IRQ_ALARM_EN0)                             | 32              | R/W    | 0000_0000h  | <a href="#">55.9.25/ 2550</a> |
| 100                  | IRQ Alarm Enable (FCCU_IRQ_ALARM_EN1)                             | 32              | R/W    | 0000_0000h  | <a href="#">55.9.25/ 2550</a> |
| 10C                  | NMI Enable (FCCU_NMI_EN0)                                         | 32              | R/W    | 0000_0000h  | <a href="#">55.9.26/ 2551</a> |
| 110                  | NMI Enable (FCCU_NMI_EN1)                                         | 32              | R/W    | 0000_0000h  | <a href="#">55.9.26/ 2551</a> |
| 11C                  | Noncritical Fault-State EOUT Signaling Enable (FCCU_EOUT_SIG_EN0) | 32              | R/W    | 0000_0000h  | <a href="#">55.9.27/ 2552</a> |
| 120                  | Noncritical Fault-State EOUT Signaling Enable (FCCU_EOUT_SIG_EN1) | 32              | R/W    | 0000_0000h  | <a href="#">55.9.27/ 2552</a> |

55.9.1 Control (FCCU\_CTRL)

Initiates and indicates the status of operations, indicates that FCCU successfully loaded configuration data from the NVMCFG signals, controls glitch-filter settings, and enables Debug mode.

Address: 0h base + 0h offset = 0h



FCCU\_CTRL field descriptions

| Field               | Description                                                                                                                                                |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>FILTER_BYPASS  | Filter bypass<br>0 glitch filter not bypassed<br>1 glitch filter bypassed                                                                                  |
| 1-2<br>FILTER_WIDTH | Filter width<br>00 filters glitches up to 50 μs<br>01 filters glitches up to 75 μs<br>10 filters glitches up to 100 μs<br>11 filters glitches up to 100 μs |

Table continues on the next page...

## FCCU\_CTRL field descriptions (continued)

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3–21<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 22<br>DEBUG      | <p>Debug Mode Enable</p> <p>Specifies whether Debug mode is enabled. If so, FCCU enters Debug mode when the Debug signal is asserted. When FCCU enters Debug mode, it halts operation and remains in the state it was in before it entered Debug mode.</p> <p><b>NOTE:</b> FOSU doesn't halt when FCCU enters Debug mode. Therefore, FOSU can still cause a reset if a fault occurs while FCCU is in Debug mode.</p> <p>0 Disabled<br/>1 Enabled</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 23<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 24–25<br>OPS     | <p>Operation status. This bit can be read and cleared (via OP15 operation) by the software.</p> <p>00 Idle<br/>01 In progress<br/>10 Aborted<br/>11 Successful</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 26<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 27–31<br>OPR     | <p>Operation Run</p> <p>Initiates operations that perform actions such as putting FCCU in Configuration state or reading a timer. For information on how to run operations, see <a href="#">Run operations</a>.</p> <p>FCCU ignores any write to this field while the operation status (<b>OPS</b>) is In Progress. After completion of an operation, FCCU sets this field to OP0.</p> <p>The following events result in an operation status (<b>OPS</b>) of Aborted:</p> <ul style="list-style-type: none"> <li>Writing to a <a href="#">Noncritical Fault Status (FCCU_NCF_Sn)</a> register (which automatically initiates the OP12 operation) without first successfully unlocking the register (<b>NCFK</b>)</li> <li>Initiating an OP1 operation when FCCU is not in Normal state or the configuration is locked</li> <li>Initiating an OP1, OP2, or OP31 operation without first unlocking the operation (<b>CTRLK</b>)</li> </ul> <p>00000 OP0—No operation</p> <p>00001 OP1—Applies only when the configuration is unlocked, when FCCU is in Normal state, and immediately after you unlock the operation (<b>CTRLK</b>). Put FCCU in Configuration state.</p> <p>00010 OP2—Applies only immediately after you unlock the operation (<b>CTRLK</b>). Put FCCU in Normal state.</p> <p>00011 OP3—Update the <a href="#">Status (FCCU_STAT)</a> register with the current states.</p> <p>00100 OP4—Update the <a href="#">NA Freeze Status (FCCU_N2AF_STATUS)</a> register with the current status.</p> <p>00101 OP5—Update the <a href="#">AF Freeze Status (FCCU_A2FF_STATUS)</a> register with the current status.</p> <p>00110 OP6—Update the <a href="#">NF Freeze Status (FCCU_N2FF_STATUS)</a> register with the current status.</p> <p>00111 OP7—Update the <a href="#">FA Freeze Status (FCCU_F2A_STATUS)</a> register with the current status.</p> <p>01000 Reserved</p> <p>01001 Reserved</p> <p>01010 OP10—Update the <a href="#">Noncritical Fault Status (FCCU_NCF_Sn)</a> registers with the current status.</p> |

Table continues on the next page...

## FCCU\_CTRL field descriptions (continued)

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | 01011 Reserved                                                                                                                                                                                                                                                                                                                                                                                                   |
|       | 01100 OP12—Do not initiate this operation; it is automatically initiated by FCCU. A <a href="#">Noncritical Fault Status (FCCU_NCF_Sn)</a> register status clear operation is in progress.                                                                                                                                                                                                                       |
|       | 01101 OP13—Clear the freeze status registers.                                                                                                                                                                                                                                                                                                                                                                    |
|       | 01110 OP14—Do not initiate this operation; it is automatically initiated by FCCU. A Configuration-state timeout is in progress. For more information, see <a href="#">Configuration registers</a> .                                                                                                                                                                                                              |
|       | 01111 OP15—Set the operation status (OPS) to Idle.                                                                                                                                                                                                                                                                                                                                                               |
|       | 10000 Reserved                                                                                                                                                                                                                                                                                                                                                                                                   |
|       | 10001 OP17—Update the <a href="#">X Timer (FCCU_XTMR)</a> register with the current value of the Alarm-state timer.                                                                                                                                                                                                                                                                                              |
|       | 10010 Reserved                                                                                                                                                                                                                                                                                                                                                                                                   |
|       | 10011 OP19—Update the <a href="#">X Timer (FCCU_XTMR)</a> register with the current value of the Configuration-state timer.                                                                                                                                                                                                                                                                                      |
|       | 10100 Reserved                                                                                                                                                                                                                                                                                                                                                                                                   |
|       | 10101–11110 Forbidden. Writing any of these values returns an operation status (OPS) of Aborted with no side effect.                                                                                                                                                                                                                                                                                             |
|       | 11111 OP31—Applies only immediately after you unlock the operation (CTRLK). Load configuration data from the NVMCFG signals to certain fields in the <a href="#">Configuration (FCCU_CFG)</a> register, but without resetting FCCU (for testing and debugging). See <a href="#">FCCU_CFG register bit value sources (N and C) by event</a> . The operation status (OPS) for this operation is always Successful. |

## 55.9.2 Control Key (FCCU\_CTRLK)

Writable only with a 32-bit write. Unlocks locked operations so you can initiate them (OPR). For information on how to unlock locked operations before you initiate them, see [Run an operation](#).

## NOTE

Reading from this register always returns the value 0000\_0000h.

Address: 0h base + 4h offset = 4h

|       |       |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0     | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |       |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     | CTRLK |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0     | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

## FCCU\_CTRLK field descriptions

| Field         | Description                  |
|---------------|------------------------------|
| 0–31<br>CTRLK | Locked-Operation Control Key |



**FCCU\_CTRLK field descriptions (continued)**

| Field | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | <p>Unlocks locked operations so you can initiate them (<a href="#">OPR</a>). For information on how to unlock locked operations before you initiate them, see <a href="#">Run an operation</a>. Operations not listed here are not locked and do not need to be unlocked.</p> <p><b>NOTE:</b> You must initiate an operation in the instruction that immediately follows the instruction that unlocks it; otherwise, the operation is again locked.</p> <p>9137_56AFh: Unlock OP1.</p> <p>825A_132Bh: Unlock OP2.</p> <p>29AF_8752h: Unlock OP31.</p> <p>Any other value: Do nothing.</p> |

**55.9.3 Configuration (FCCU\_CFG)**

Writable only when FCCU is in Configuration state. Specifies the global configuration for FCCU.

The values of the bits in this register are affected by writes as well as any of the following events:

- The chip resets FCCU.
- FCCU automatically leaves Configuration state and enters Normal state because the configuration-timer interval ([TO](#)) expires. This is called a Configuration-state timeout.
- You run an OP31 operation.

For information on the sources of bit values for these events, see [FCCU\\_CFG register bit value sources \(N and C\) by event](#).

**NOTE**

If you specify a new value for any of the fields in this register that affect the EOUT signals while the fault-output (EOUT) timer is running (FCCU is indicating a fault on the EOUT signals), FCCU doesn't use the new settings you specified until after the fault-output (EOUT) timer expires (FCCU stops indicating a fault on the EOUT signals).

## Register descriptions

Address: 0h base + 8h offset = 8h

| Bit   | 0        | 1  | 2  | 3  | 4  | 5  | 6  | 7                    | 8              | 9  | 10       | 11 | 12       | 13 | 14 | 15 |
|-------|----------|----|----|----|----|----|----|----------------------|----------------|----|----------|----|----------|----|----|----|
| R     | Reserved |    |    |    |    |    |    | FCCU_SET_AFTER_RESET | FCCU_SET_CLEAR |    | Reserved |    | Reserved |    |    |    |
| W     |          |    |    |    |    |    |    |                      |                |    |          |    |          |    |    |    |
| Reset | x*       | x* | x* | x* | x* | x* | x* | x*                   | x*             | x* | x*       | x* | x*       | x* | x* | x* |

| Bit   | 16   | 17       | 18 | 19       | 20       | 21 | 22 | 23  | 24  | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|-------|------|----------|----|----------|----------|----|----|-----|-----|----|----|----|----|----|----|----|
| R     | FOPE | Reserved |    | Reserved | Reserved | SM | PS | FOM | FOP |    |    |    |    |    |    |    |
| W     |      |          |    |          |          |    |    |     |     |    |    |    |    |    |    |    |
| Reset | x*   | x*       | x* | x*       | x*       | x* | x* | x*  | x*  | x* | x* | x* | x* | x* | x* | x* |

\* Notes:

- x = Undefined at reset.

## FCCU\_CFG field descriptions

| Field                     | Description                                                                                                      |
|---------------------------|------------------------------------------------------------------------------------------------------------------|
| 0–6<br>Reserved           | This field is reserved.                                                                                          |
| 7<br>FCCU_SET_AFTER_RESET | Fault-Output (EOUT) Activate<br>For fault-output (EOUT) signaling, controls whether the EOUT signals are active. |

Table continues on the next page...

**FCCU\_CFG field descriptions (continued)**

| Field                  | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                        | 0 Inactive (the EOUT signals are in a high-impedance state)<br>1 Active (the EOUT signals indicate FCCU's condition)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 8–9<br>FCCU_SET_ CLEAR | Fault-Output (EOUT) Control<br><br>Applies only when the EOUT signals are active ( <a href="#">FCCU_SET_AFTER_RESET</a> ). Controls whether the fault-output (EOUT) signals are controlled by FCCU's finite state machine (FSM).<br><br>00 Controlled by the FSM<br>01 Always low<br>10 Controlled by the FSM<br>11 High until a fault occurs on a channel, regardless of whether that fault is disabled; thereafter, controlled by the FSM. Note: FCCU ignores an attempt to write this value if the fault-output (EOUT) timer is already running.                                                                                                                                                                                                                       |
| 10–11<br>Reserved      | This field is reserved.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 12–15<br>Reserved      | This field is reserved.<br>Always write the reset value to this field.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 16<br>FOPE             | Fault-Output (EOUT) Prescaler Extension<br><br>For fault-output (EOUT) signaling, specifies the most significant bit of the derived prescaler that controls the signaling frequency. For more information, see the description of the FOP field.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 17–18<br>Reserved      | This field is reserved.<br>Always write the reset value to this field.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 19<br>Reserved         | This field is reserved.<br>Always write the reset value to this field.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 20<br>Reserved         | This field is reserved.<br>Always write the reset value to this field.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 21<br>SM               | Fault-Output (EOUT) Switching Mode<br><br>For fault-output (EOUT) signaling, controls how quickly the signals switch between faulty and nonfaulty indication. Applies only to Dual-Rail and Time-Switching fault-output modes.<br><br>0 Slow: No EOUT frequency violation during the FCCU state transition (Normal to Fault or vice versa, and Configuration to Normal). The indication transition occurs after a maximum delay equal to the duration of the semiperiod of the EOUT frequency.<br>1 Fast: The indication transition (Normal to Fault or vice versa, and Configuration to Normal) occurs immediately. A pulse with the minimum duration corresponding to the CLKSAFE period can occur in this mode. It implies a frequency violation of the EOUT protocol. |
| 22<br>PS               | Fault-Output (EOUT) Polarity Selection<br><br>For fault-output (EOUT) signaling, controls the polarity of the signals for fault-output-mode indications that hold the signals low or high (versus toggling them or placing them in a high-impedance state). Applies only to Time-Switching fault-output mode (for faulty and configuration indications) and Bistable fault-output mode (for all indications).<br><br>0 For the faulty indication, EOUT1 is high, and EOUT0 is low.<br>1 For the faulty indication, EOUT1 is low, and EOUT0 is high.                                                                                                                                                                                                                       |
| 23–25<br>FOM           | Fault-Output (EOUT) Mode<br><br>For fault-output (EOUT) signaling, controls the protocol of the signaling.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

*Table continues on the next page...*

## FCCU\_CFG field descriptions (continued)

| Field        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|              | <p><b>NOTE:</b> In the test modes, a simple double-stage resynchronization stage is used to resynchronize the EOUT input/outputs with CLKSAFE.</p> <p>000 Dual-Rail (default state) [EOUT[1:0]= outputs]<br/> 001 Time-Switching [EOUT[1:0]= output to be used]<br/> 010 Bistable<br/> 011 Reserved<br/> 100 Reserved<br/> 101 Test 0 (controlled by the FCCU_EINOUT register; EOUT1 is an output; EOUT0 is an input)<br/> 110 Test 1 (controlled by the FCCU_EINOUT register; EOUT1 and EOUT0 are both outputs)<br/> 111 Test 2 (controlled by the FCCU_EINOUT register; EOUT1 is an input; EOUT0 is an output)</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 26–31<br>FOP | <p>Fault-Output (EOUT) Prescaler</p> <p>For fault-output (EOUT) signaling, specifies the least significant bits of the derived prescaler that controls the signaling frequency.</p> <p>The derived prescaler is the concatenation of FOPE and FOP and is represented by <math>\{FOPE, FOP\}</math>.</p> <p>The following equation defines the signaling frequency in terms of the CLKSAFE frequency (<math>CLKSAFE_{freq}</math>) and the derived prescaler:</p> $EOUT_{freq} = CLKSAFE_{freq} / ((\{FOPE, FOP\} + 1) \times 2 \times 2048)$ <p>The following example values and meanings are for the derived (7-bit) prescaler, not FOP alone. The meanings show the derived prescaler followed by the corresponding value by which the input clock frequency is then divided.</p> <p>0000000: 0; divide by <math>2 \times 2048</math> (4096)<br/> 0000001: 1; divide by <math>4 \times 2048</math> (8192)<br/> 0000010: 2; divide by <math>6 \times 2048</math> (12,288)<br/> ...<br/> 0001000: 8; divide by <math>18 \times 2048</math> (36,864)<br/> ...<br/> 1111101: 125; divide by <math>252 \times 2048</math> (516,096)<br/> 1111110: 126; divide by <math>254 \times 2048</math> (520,192)<br/> 1111111: 127; divide by <math>256 \times 2048</math> (524,288)</p> <p><b>NOTE:</b> Writing to FCCU_CFG sets FOP to 0.</p> |

## 55.9.4 Noncritical Fault Configuration (FCCU\_NCF\_CFGn)

Contains the configuration of noncritical faults in terms of fault-recovery management. The configuration depends on the type of signaling of a fault event. Hardware-recoverable faults should be configured only if a previous latching stage captures and holds the physical fault; otherwise, the fault can be lost. All other faults should be configured as software faults.

**NOTE**

These registers are writable only when FCCU is in Configuration state.

| Noncritical Fault Configuration<br>(FCCU_NCF_CFGn)<br>register (value of n) | Offset | NCFCx fields (value of x)       |                                   |
|-----------------------------------------------------------------------------|--------|---------------------------------|-----------------------------------|
|                                                                             |        | Most significant (leftmost) bit | Least significant (rightmost) bit |
| 0                                                                           | 1Ch    | 31                              | 0                                 |
| 1                                                                           | 20h    | 63                              | 32                                |

Address: 0h base + 1Ch offset + (4d × i), where i=0d to 1d

|       |       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0     | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     | NCFCx |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*    | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

**FCCU\_NCF\_CFGn field descriptions**

| Field         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>NCFCx | <p>Noncritical Fault Configuration. Defines the fault recovery mode.</p> <p>Hardware-recoverable faults are self-recovered (status flag clearing and related) if the root cause (input fault) has been removed.</p> <p>Software-recoverable faults are recovered (status flag clearing) by software clearing of the related status flag.</p> <p>For the mapping of the NCFCx fields among registers, see the earlier table in this register description.</p> <p>0 Hardware-recoverable fault</p> <p>1 Software-recoverable fault</p> |

**55.9.5 Noncritical Fault State Configuration (FCCU\_NCFS\_CFGn)**

The FCCU\_NCFS\_CFGx registers contain the configuration of each noncritical fault in terms of fault reaction (functional reset request pulse) when it is the root cause for the FAULT state transition.

**NOTE**

These registers are writable only when the FCCU is in the CONFIG state.

Table 55-16 shows FCCU configuration register channels.

**Table 55-18. FCCU NCFS configuration register channels**

| Address offset | Register name  | Channel range (x) |
|----------------|----------------|-------------------|
| 4Ch            | FCCU_NCFS_CFG0 | NCFSCx[15:0]      |
| 50h            | FCCU_NCFS_CFG1 | NCFSCx[31:16]     |
| 54h            | FCCU_NCFS_CFG2 | NCFSCx[47:32]     |

Address: 0h base + 4Ch offset + (4d × i), where i=0d to 2d

|       |        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|--------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0      | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     | NCFSCx |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | X*     | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* |    |

\* Notes:

- x = Undefined at reset.

### FCCU\_NCFS\_CFGn field descriptions

| Field          | Description                                                                                                                                                                                                                                                                        |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>NCFSCx | Noncritical fault state configuration<br>See <a href="#">Table 55-16</a> for register offset to channel number relationship.<br><br>00 No reset reaction<br>01 Reset request pulse (FAULT state reaction)<br>10 Reset request pulse (FAULT state reaction)<br>11 No reset reaction |

## 55.9.6 Noncritical Fault Status (FCCU\_NCF\_Sn)

The FCCU\_NCF\_Sx register contains the latched fault indication collected from the noncritical fault sources. Faults are latched also in the CONFIG state and independently from the enabling or reactions programmed for the NCF.

No reactions are executed until the FCCU moves in the NORMAL state.

FCCU reacts and moves from the NORMAL state into the ALARM state only if the respective enable bit for a fault is set in the FCCU\_NCF\_Ex register and the respective enable bit for the timeout is set in the FCCU\_TOEx register.

FCCU reacts and moves from the NORMAL or ALARM state into the FAULT state if the respective enable bit for a fault is set in the FCCU\_NCF\_Ex register and the respective enable bit for the timeout is disabled in the FCCU\_TOEx register.

FCCU reacts and moves from the ALARM state into the FAULT state if the timeout (FCCU\_TO register) is elapsed before recovering from the fault.

The timeout is stopped only when the FCCU returns in the NORMAL state.

The FCCU\_NCF\_Sx register is encoded respectively into the N2FF\_STATUS or A2FF\_STATUS register to freeze the entry condition in the FAULT state. The status bits of the FCCU\_NCF\_Sx register, configured as SW recoverable faults, can be cleared by the following locked sequence:

1. Write the proper key into the FCCU\_NCFK register.
2. Clear the status (flag) bit NCFSx => the opcode OP12 is automatically set into the FCCU\_CTRL.OPR field.
3. Wait for the completion of the operation (FCCU\_CTRL.OPS field).
4. Read the FCCU\_NCF\_Sx register in order to verify the effective deletion and in case of failure to repeat the sequence.

As result of the above sequence, in addition the FAULT interface provides support to clear the external fake FAULT root .

### NOTE

There should not be any other operation in between the above steps.

The FCCU moves from the FAULT or ALARM state into the NORMAL state if all the source faults that caused the transition into the FAULT state have been removed (HW recoverable fault) or cleared via SW (SW recoverable fault). In case of nested faults that are not all recovered, the FCCU will remain in the FAULT or ALARM state.

The SW application executes the FCCU\_NCF\_Sx read operation by the following sequence:

1. Set the OP10 operation into the FCCU\_CTRL.OPR field.
2. Wait for the completion of the operation (FCCU\_CTRL.OPS field).
3. Read the FCCU\_NCF\_Sx register.

In case of re-configuration of the FCCU (CONFIG state), before to return in NORMAL state the pending status bits into the FCCU\_NCF\_Sx must be cleared in order to avoid a false transition in ALARM/FAULT state.

The following faults are ignored:

- to write a wrong key into the FCCU\_NCFK register
- to attempt to clear a HW recoverable fault

| Noncritical Fault<br>Status<br>(FCCU_NCF_Sn)<br>register (value of n) | Offset | NCFSx fields (value of x)          |                                      |
|-----------------------------------------------------------------------|--------|------------------------------------|--------------------------------------|
|                                                                       |        | Most significant (leftmost)<br>bit | Least significant<br>(rightmost) bit |
| 0                                                                     | 80h    | 31                                 | 0                                    |
| 1                                                                     | 84h    | 63                                 | 32                                   |

## Register descriptions

Address: 0h base + 80h offset + (4d × i), where i=0d to 1d

|       |       |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0     | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | NCFSx |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     | w1c   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0     | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

### FCCU\_NCF\_Sn field descriptions

| Field         | Description                                                                                                                                                                                                                                                                                                                                                                                                  |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>NCFSx | <p>Noncritical fault status. The status bits related to the noncritical fault configured as HW recoverable faults are read-only and the flag is self cleared when the fault source is removed.</p> <p>For the mapping of the <a href="#">NCFSx</a> fields among registers, see the earlier table in this register description.</p> <p>0 No "noncritical" fault latched<br/>1 "Noncritical" fault latched</p> |

## 55.9.7 Noncritical Fault Key (FCCU\_NCFK)

The FCCU\_NCFK register implements the key access to clear the status flags of the FCCU\_NCF\_Sx register.

The status bits of the FCCU\_NCF\_Sx register, configured as SW recoverable faults, can be cleared by the following locked sequence:

1. Write the key into the FCCU\_NCFK register.
2. Clear the status (flag) bit NCFSx .

The key must be written for each FCCU\_NCF\_Sx clear operation.

The FCCU\_NCFK register is not readable; a 0000\_0000h value is always returned in case of read operation.

Address: 0h base + 90h offset = 90h

|       |      |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0    |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     | NCFK |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0    | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

### FCCU\_NCFK field descriptions

| Field        | Description                        |
|--------------|------------------------------------|
| 0–31<br>NCFK | Noncritical fault key = AB34_98FEh |



## 55.9.8 Noncritical Fault Enable (FCCU\_NCF\_En)

The FCCU\_NCF\_En registers enable the fault sources to allow a transition from the NORMAL into the FAULT or ALARM state. In case of fault masking, the respective status bit into the FCCU\_NCF\_Sn register is set (for debugging purposes), only the reaction is masked.

### NOTE

These registers are writable only when the FCCU is in the CONFIG state.

### NOTE

Any enabled fault should be programmed to result in a defined action. For example, set up an ALARM IRQ action by programming the IRQ Alarm Enable Register (FCCU\_IRQ\_ALARM\_ENn).

| Noncritical Fault Enable<br>(FCCU_NCF_En)<br>register (value of n) | Offset | NCFEx fields (value of x)       |                                   |
|--------------------------------------------------------------------|--------|---------------------------------|-----------------------------------|
|                                                                    |        | Most significant (leftmost) bit | Least significant (rightmost) bit |
| 0                                                                  | 94h    | 31                              | 0                                 |
| 1                                                                  | 98h    | 63                              | 32                                |

Address: 0h base + 94h offset + (4d × i), where i=0d to 1d

|       |       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0     | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     | NCFEx |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | X*    | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* |    |

\* Notes:

- x = Undefined at reset.

### FCCU\_NCF\_En field descriptions

| Field         | Description                                                                                                                                                                                                                                       |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>NCFEx | Noncritical fault enable<br>For the mapping of the NCFEx fields among registers, see the earlier table in this register description.<br>0 No actions following the respective noncritical fault assertion<br>1 FCCU moves to ALARM or FAULT state |

### 55.9.9 Noncritical Fault Timeout Enable (FCCU\_NCF\_TOEn)

The FCCU\_NCF\_TOEx registers enable a transition from the NORMAL state into the ALARM state if the respective noncritical fault is enabled ( NCFEx and NCFTOEx are set). In case the respective timeout is disabled ( NCFTOEx is cleared) and the noncritical fault is enabled (NCFEx is set) the FCCU moves into the FAULT state if the related noncritical fault is asserted. The timer (preset with the timeout value defined by FCCU\_TO register) is started when the FCCU moves into the ALARM state. If the fault is not recovered within the timeout the FCCU moves from the ALARM state to the FAULT state.

#### NOTE

These registers are writable only when the FCCU is in the CONFIG state.

| Noncritical Fault Timeout Enable (FCCU_NCF_TOEn) register (value of n) | Offset | NCFTOEx fields (value of x)     |                                   |
|------------------------------------------------------------------------|--------|---------------------------------|-----------------------------------|
|                                                                        |        | Most significant (leftmost) bit | Least significant (rightmost) bit |
| 0                                                                      | A4h    | 31                              | 0                                 |
| 1                                                                      | A8h    | 63                              | 32                                |

Address: 0h base + A4h offset + (4d × i), where i=0d to 1d

|       |         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0       | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     | NCFTOEx |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*      | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |

\* Notes:

- x = Undefined at reset.

#### FCCU\_NCF\_TOEn field descriptions

| Field           | Description                                                                                                                                                                                                                                                                                           |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>NCFTOEx | <p>Fault timeout enable</p> <p>For the mapping of the NCFTOEx fields among registers, see the earlier table in this register description.</p> <p>0 FCCU moves into the FAULT state if the respective fault is enabled</p> <p>1 FCCU moves into the ALARM state if the respective fault is enabled</p> |

### 55.9.10 Noncritical Fault Timeout (FCCU\_NCF\_TO)

Defines the preset value of the timer for the recovery of enabled noncritical faults. Once FCCU enters Alarm state, following the assertion of a noncritical fault that is enabled (NCFEa and NCFTOEa are set), the timer starts the countdown.

If the fault is not recovered within the timeout period, the FCCU moves from the Alarm state to the Fault state. The alarm timeout value should be programmed to be less than FOSU\_COUNT, or resets may be generated by FOSU timeouts.

#### NOTE

This register is writable only when FCCU is in Configuration state.

Address: 0h base + B4h offset = B4h

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | TO |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

#### FCCU\_NCF\_TO field descriptions

| Field      | Description                                                        |
|------------|--------------------------------------------------------------------|
| 0–31<br>TO | Noncritical Fault Timeout<br>Timeout = (TO) × T <sub>RC16MHz</sub> |

### 55.9.11 Configuration-State Timer Interval (FCCU\_CFG\_TO)

Writable only when FCCU is in Normal, Alarm, or Fault state (not in Configuration state). Not accessible while a Configuration-state timeout (OP14 operation) is in progress. Controls the maximum amount of time that FCCU can be in Configuration state.

#### NOTE

When a Configuration-state timeout occurs, FCCU changes the value of this register to its reset value.

## Register descriptions

Address: 0h base + B8h offset = B8h

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | TO |    |    |    |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

## FCCU\_CFG\_TO field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–28<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 29–31<br>TO      | <p>Configuration-State Timer Interval<br/>Controls the maximum amount of time (T<sub>max</sub>) that FCCU can be in Configuration state according to this equation:</p> $T_{\text{max}} = T_{16\text{MHz}} \times 2^{(\text{TO} + 10)}$ <p>If you put FCCU in Configuration state and this timer interval expires (called a Configuration-state timeout), then FCCU:</p> <ul style="list-style-type: none"> <li>• Automatically leaves Configuration state and enters Normal state.</li> <li>• Changes the value of the <a href="#">Configuration (FCCU_CFG)</a> register to its Configuration-state-timeout value and the value of each of the other configuration registers to its reset value. For information on the <a href="#">Configuration (FCCU_CFG)</a> register's Configuration-state-timeout value, see <a href="#">FCCU_CFG register bit value sources (N and C) by event</a>. For a list of configuration registers, see <a href="#">Configuration registers</a>.</li> </ul> <p>000: 64 μs<br/>...<br/>111: 8192 μs</p> |

## 55.9.12 IO Control (FCCU\_EINOUT)

The FCCU\_EINOUT register allows the following operations typically in NORMAL state:

- to control the EOUT[1] output level when the FCCU is configured in "Test1" or "Test0" fault output mode (FCCU\_CFG.FOM)
- to control the EOUT[0] output level when the FCCU is configured in "Test1" or "Test2" fault output mode (FCCU\_CFG.FOM)
- to observe the EOUT[1:0] signals in input mode

[Table 55-36](#) shows Bi-stable encoding.

**Table 55-36. Bi-stable encoding**

| Mode = FCCU_CFG.FOM | EOUT[0] | EOUT[1] |
|---------------------|---------|---------|
| Test1               | output  | output  |
| Test2               | output  | input   |
| Test0               | input   | output  |

**NOTE**

Due to the resynchronization stage of the EOUT interface, there is a latency of a few CLKSAFE cycles following a write/read operation of the FCCU\_EINOUT register.

Address: 0h base + BCh offset = BCh

|       |    |    |    |    |    |    |    |    |    |    |      |      |    |    |       |       |
|-------|----|----|----|----|----|----|----|----|----|----|------|------|----|----|-------|-------|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10   | 11   | 12 | 13 | 14    | 15    |
| R     | 0  |    |    |    |    |    |    |    |    |    |      |      |    |    |       |       |
| W     |    |    |    |    |    |    |    |    |    |    |      |      |    |    |       |       |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0    | 0  | 0  | 0     | 0     |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26   | 27   | 28 | 29 | 30    | 31    |
| R     | 0  |    |    |    |    |    |    |    |    |    | EIN1 | EIN0 | 0  |    | EOUT1 | EOUT0 |
| W     |    |    |    |    |    |    |    |    |    |    |      |      |    |    |       |       |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | x*   | x*   | 0  | 0  | 0     | 0     |

\* Notes:

- x = Undefined at reset.

**FCCU\_EINOUT field descriptions**

| Field            | Description                                                                             |
|------------------|-----------------------------------------------------------------------------------------|
| 0–25<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |

Table continues on the next page...

**FCCU\_EINOUT field descriptions (continued)**

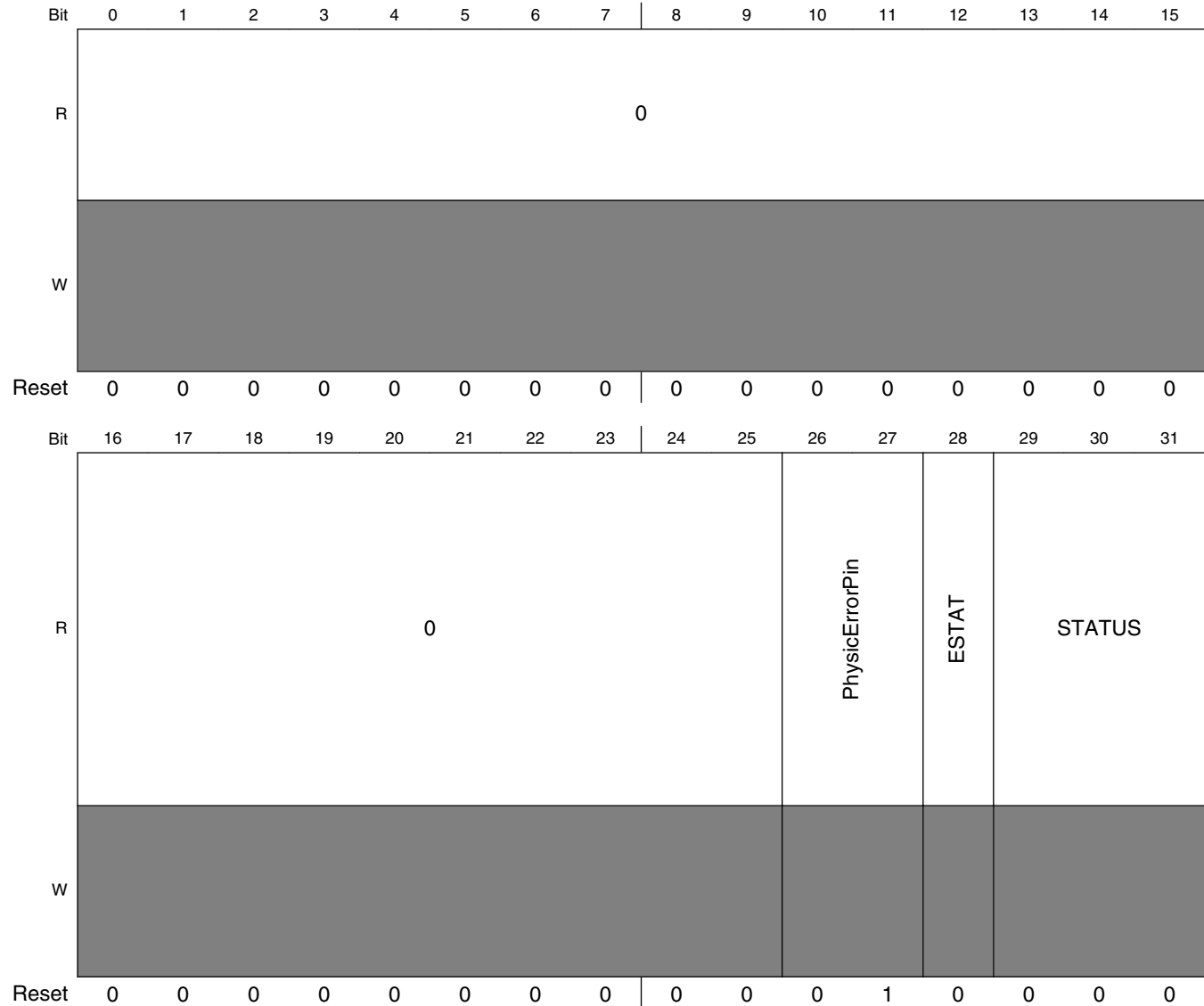
| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 26<br>EIN1        | <p>Error Input 1</p> <p>Applies only when the EOUT signals are active (<a href="#">FCCU_SET_AFTER_RESET</a>). Indicates the state of the EIN1 signal.</p> <p>0 Low<br/>1 High</p>                                                                                                                                                                                                                                               |
| 27<br>EIN0        | <p>Error Input 0</p> <p>Applies only when the EOUT signals are active (<a href="#">FCCU_SET_AFTER_RESET</a>). Indicates the state of the EIN0 signal.</p> <p>0 Low<br/>1 High</p>                                                                                                                                                                                                                                               |
| 28–29<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                              |
| 30<br>EOUT1       | <p>Error out 1 (significant only if the FCCU_CFG.FOM = Test1 or Test0 =&gt; EOUT[1] configured in output mode). The EOUT1 set/clear the respective EOUT[1] output signal if FCCU_CFG.FOM = 110 or 101, otherwise it is a "don't-care" value.</p> <p><b>NOTE:</b> When the configuration watchdog timer expires, FCCU changes the value of this field to its reset value.</p> <p>0 force EOUT[1] = 0<br/>1 force EOUT[1] = 1</p> |
| 31<br>EOUT0       | <p>Error out 0 (significant only if the FCCU_CFG.FOM = Test1 or Test2 =&gt; EOUT[0] configured in output mode). The EOUT0 set/clear the respective EOUT[0] output signal if FCCU_CFG.FOM = 110 or 111, otherwise it is a "don't care" value.</p> <p><b>NOTE:</b> When the configuration watchdog timer expires, FCCU changes the value of this field to its reset value.</p> <p>0 force EOUT[0] = 0<br/>1 force EOUT[0] = 1</p> |

**55.9.13 Status (FCCU\_STAT)**

The FCCU\_STAT register includes the FCCU status for debugging/test purposes. The FCCU finite state machine operates by the RC oscillator clock asynchronous with the system clock. The FCCU status read operation requires a safe mechanism operated by a HW/SW synchronization sequence. The SW application executes a FCCU status read operation by the following sequence:

1. Set the OP3 operation into the FCCU\_CTRL.OPR field.
2. Wait for the completion of the operation (FCCU\_CTRL.OPS field).
3. Read the FCCU status (FCCU\_STAT register).

Address: 0h base + C0h offset = C0h

**FCCU\_STAT field descriptions**

| Field                   | Description                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–25<br>Reserved        | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                               |
| 26–27<br>PhysicErrorPin | Fault Output (EOUT) States<br><br>Applies only after successfully running an OP3 operation ( <a href="#">OPR</a> ) and only when the EOUT signals are active ( <a href="#">FCCU_SET_AFTER_RESET</a> ). Indicates the states that FCCU is driving on the EOUT signals.<br><br>00 EOUT1 is low; EOUT0 is low.<br>01 EOUT1 is low; EOUT0 is high.<br>10 EOUT1 is high; EOUT0 is low.<br>11 EOUT1 is high; EOUT0 is high. |
| 28<br>ESTAT             | Fault State                                                                                                                                                                                                                                                                                                                                                                                                           |

*Table continues on the next page...*

**FCCU\_STAT field descriptions (continued)**

| Field           | Description                                                                                               |
|-----------------|-----------------------------------------------------------------------------------------------------------|
|                 | Applies only after successfully running an OP3 operation (OPR). Indicates whether FCCU is in Fault state. |
|                 | 0 Not in Fault state (in Normal, Alarm, or Configuration state)                                           |
|                 | 1 In Fault state                                                                                          |
| 29–31<br>STATUS | FCCU State<br><br>Applies only after successfully running an OP3 operation (OPR). Indicates FCCU's state. |
|                 | 000 Normal                                                                                                |
|                 | 001 Configuration                                                                                         |
|                 | 010 Alarm                                                                                                 |
|                 | 011 Fault                                                                                                 |
|                 | 100 Reserved                                                                                              |
|                 | 101 Reserved                                                                                              |
|                 | 110 Reserved                                                                                              |
|                 | 111 Reserved                                                                                              |

**55.9.14 NA Freeze Status (FCCU\_N2AF\_STATUS)**

The N2AF\_STATUS register contains a unique code to identify the "noncritical fault" source ( fccu\_ncf[x] ) that caused the state transition from the NORMAL state to the ALARM state. In case of multiple fault conditions the fault source cannot be identified. This register is for test/debug purposes.

The SW application executes the N2AF\_STATUS read operation by the following sequence:

1. Set the OP4 operation into the FCCU\_CTRL.OPR field.
2. Wait for the completion of the operation (FCCU\_CTRL.OPS field).
3. Read the N2AF\_STATUS register.

The SW application executes the N2AF\_STATUS clear operation by the following sequence:

1. Set the OP13 operation into the FCCU\_CTRL.OPR field.
2. Wait for the completion of the operation (FCCU\_CTRL.OPS field). (All the freeze registers are cleared by this operation.)

Address: 0h base + C4h offset = C4h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23   | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    | NAFS |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |



**FCCU\_N2AF\_STATUS field descriptions**

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–23<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                        |
| 24–31<br>NAFS    | Normal to Alarm Frozen Status<br>00000000 No transition from NORMAL to ALARM state<br>00000001 NORMAL to ALARM state transition cause => fccu_ncf[0] fault<br>00000010 NORMAL to ALARM state transition cause => fccu_ncf[1] fault<br>00000011 NORMAL to ALARM state transition cause => fccu_ncf[2] fault<br>...<br>10000000 NORMAL to ALARM state transition cause => fccu_ncf[127]/ fault<br>11111111 NORMAL to ALARM state transition cause => multiple fccu_ncf[]/ faults |

**55.9.15 AF Freeze Status (FCCU\_A2FF\_STATUS)**

The A2FF\_STATUS register contains a unique code to identify the timeout trigger (noncritical fault) that caused the state transition from the ALARM state to the FAULT state. In case of multiple fault conditions the fault source cannot be identified. This register is for test/debug purposes.

The SW application executes the A2FF\_STATUS read operation by the following sequence:

1. Set the OP5 operation into the FCCU\_CTRL.OPR field.
2. Wait for the completion of the operation (FCCU\_CTRL.OPS field).
3. Read the A2FF\_STATUS register.

The SW application executes the FCCU\_AFFS clear operation by the following sequence:

1. Set the OP13 operation into the FCCU\_CTRL.OPR field.
2. Wait for the completion of the operation (FCCU\_CTRL.OPS field).

All the freeze registers are cleared by this operation.

Address: 0h base + C8h offset = C8h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |        |    |      |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|--------|----|------|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21     | 22 | 23   | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    | AF_SRC |    | AFFS |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |        |    |      |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0      | 0  | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**FCCU\_A2FF\_STATUS field descriptions**

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–21<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 22–23<br>AF_SRC  | Fault source<br>00 No fault<br>01 Reserved<br>10 "Noncritical" fault<br>11 Multiple and/or "noncritical" faults                                                                                                                                                                                                                                                                                                                                                                                                           |
| 24–31<br>AFFS    | Alarm to Fault Frozen Status<br>00h: No transition from ALARM to FAULT state<br>01h: ALARM to FAULT state transition cause => fccu_ncf[0]/ fault timeout fault<br>02h: ALARM to FAULT state transition cause => fccu_ncf[1]/ fault timeout fault<br>03h: ALARM to FAULT state transition cause => fccu_ncf[2]/ fault timeout fault<br>...<br>80h: ALARM to FAULT state transition cause => fccu_ncf[127]/ fault timeout fault<br>FFh: ALARM to FAULT state transition cause => multiple fccu_ncf[]/ faults timeout faults |

**55.9.16 NF Freeze Status (FCCU\_N2FF\_STATUS)**

The N2FF\_STATUS register contains a unique code to identify the source of the noncritical fault that caused the state transition from the NORMAL state to the FAULT state. In case of multiple fault conditions the fault source cannot be identified. This register is for test/debug purposes. The SW application executes the N2FF\_STATUS read operation by the following sequence:

1. Set the OP6 operation into the FCCU\_CTRL.OPR field.
2. Wait for the completion of the operation (FCCU\_CTRL.OPS field).
3. Read the N2FF\_STATUS register.

The SW application executes the N2FF\_STATUS clear operation by the following sequence:

1. Set the OP13 operation into the FCCU\_CTRL.OPR field.
2. Wait for the completion of the operation (FCCU\_CTRL.OPS field). (All the freeze registers are cleared by this operation.)

Address: 0h base + CCh offset = CCh

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |        |    |      |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|--------|----|------|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21     | 22 | 23   | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    | NF_SRC |    | NFFS |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |        |    |      |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0      | 0  | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

### FCCU\_N2FF\_STATUS field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–21<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                              |
| 22–23<br>NF_SRC  | NF_SRC: Fault source<br><br>00 No fault<br>01 Reserved<br>10 "Noncritical" fault<br>11 Multiple "noncritical" faults                                                                                                                                                                                                                                                                                                                                 |
| 24–31<br>NFFS    | Normal to Fault Frozen Status<br>00h: No transition from NORMAL to FAULT state<br>01h: NORMAL to FAULT state transition cause => fccu_ncf[0] fault<br>02h: NORMAL to FAULT state transition cause => fccu_ncf[1] fault<br>03h: NORMAL to FAULT state transition cause => fccu_ncf[2] fault<br>...<br>80h: NORMAL to FAULT state transition cause => fccu_ncf[127] fault<br>FFh: NORMAL to FAULT state transition cause => multiple fccu_ncf[] faults |

## 55.9.17 FA Freeze Status (FCCU\_F2A\_STATUS)

The F2AF\_STATUS register contains a unique code to identify the source of the noncritical fault (fccu\_ncf[x]) that caused the state transition from the FAULT state to the ALARM state. In case of multiple fault conditions the fault source cannot be identified. This register is for test/debug purposes.

The SW application executes the F2AF\_STATUS read operation by the following sequence:

1. Set the OP7 operation into the FCCU\_CTRL.OPR field.
2. Wait for the completion of the operation (FCCU\_CTRL.OPS field).
3. Read the F2AF\_STATUS register.

The SW application executes the F2AF\_STATUS clear operation by the following sequence:

1. Set the OP13 operation into the FCCU\_CTRL.OPR field.

## Register descriptions

- Wait for the completion of the operation (FCCU\_CTRL.OPS field). (All the freeze registers are cleared by this operation.)

Address: 0h base + D0h offset = D0h

| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22   | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|------|----|----|----|----|----|----|----|----|----|
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    | FAFS |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

### FCCU\_F2A\_STATUS field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–22<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                            |
| 23–31<br>FAFS    | Fault to Normal Frozen Status<br>00h: No transition from FAULT to ALARM state<br>01h: FAULT to ALARM state transition cause => fccu_ncf[0]/ fault<br>02h: FAULT to ALARM state transition cause => fccu_ncf[1]/ fault<br>03h: FAULT to ALARM state transition cause => fccu_ncf[2]/ fault<br>...<br>80h: FAULT to ALARM state transition cause => fccu_ncf[127]/ fault<br>FFh: FAULT to ALARM state transition cause => multiple fccu_ncf[] faults |

## 55.9.18 Noncritical Fault Fake (FCCU\_NCFF)

The FCCU\_NCFF register contains a unique code to set a noncritical fault in mutually exclusive mode by the external FAULT interface (signal setting). It allows the SW emulation of the noncritical faults, by the injection of the fault directly in the FAULT root, in order to verify the entire path and reaction. The reaction following a fake noncritical fault cannot be masked. The FCCU\_NCFF is a write-only register with a set of codes corresponding to each noncritical fault injection.

Address: 0h base + DCh offset = DCh

| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24    | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|----|----|----|----|----|----|----|
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    | FNCFC |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0     | 0  | 0  | 0  | 0  | 0  | 0  |    |

**FCCU\_NCFE field descriptions**

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–24<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                               |
| 25–31<br>FNCFC   | FNCFC/ : Fake noncritical fault code<br><br><b>NOTE:</b> Only writing to this register, fake fault injection occurs, writing 00 and default value being zero give different results.<br><br>00h: Fake noncriticalfault injection at noncriticalfault source 0<br>01h: Fake noncriticalfault injection at noncriticalfault source 1<br>02h: Fake noncriticalfault injection at noncriticalfault source 2<br>...<br>7Fh: Fake noncriticalfault injection at noncriticalfault source 127 |

**55.9.19 IRQ Status (FCCU\_IRQ\_STAT)**

The FCCU\_IRQ\_STAT register provides the FCCU interrupt status related to the following events:

- Configuration timeout error
- Alarm interrupt
- NMI interrupt

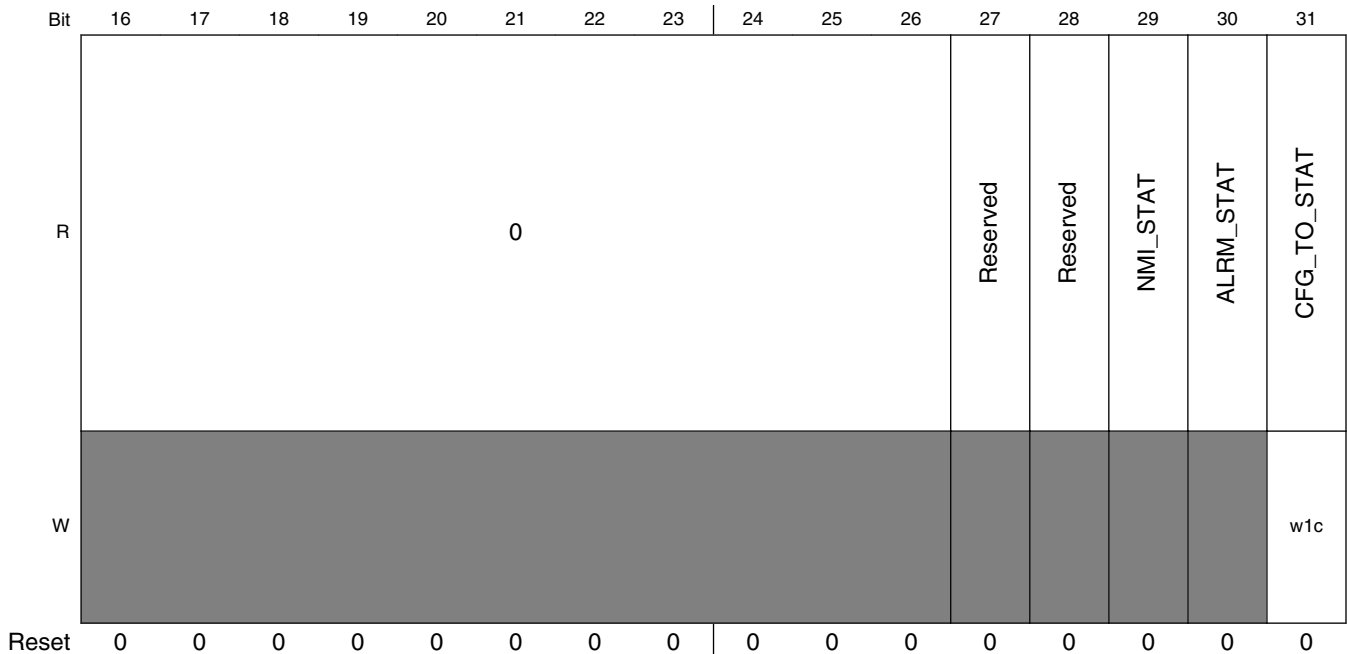
The configuration timeout interrupt is asserted if the CFG\_TO\_STAT bit of the FCCU\_IRQ\_STAT register is set and the CFG\_TO\_IEN bit of the FCCU\_IRQ\_EN register is also set. It is cleared when a 1 is written to the CFG\_TO\_STAT bit.

The NMI and ALARM interrupts are asserted and cleared according to the FCCU state. The status bits of the FCCU\_IRQ\_STAT trace the status of the related interrupt lines.

Address: 0h base + E0h offset = E0h

| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| R     |   |   |   |   |   |   |   |   | 0 |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

Register descriptions



FCCU\_IRQ\_STAT field descriptions

| Field             | Description                                                                                       |
|-------------------|---------------------------------------------------------------------------------------------------|
| 0–26<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.           |
| 27<br>Reserved    | This field is reserved.                                                                           |
| 28<br>Reserved    | This field is reserved.                                                                           |
| 29<br>NMI_STAT    | NMI Interrupt Status<br>0 NMI interrupt is OFF<br>1 NMI interrupt is ON                           |
| 30<br>ALRM_STAT   | Alarm Interrupt Status<br>0 Alarm interrupt is OFF<br>1 Alarm interrupt is ON                     |
| 31<br>CFG_TO_STAT | Configuration Timeout Status<br>0 No configuration timeout error<br>1 Configuration timeout error |

55.9.20 IRQ Enable (FCCU\_IRQ\_EN)

The FCCU\_IRQ\_EN register defines the FCCU interrupt enable register related to the following event:

- Configuration timeout error

The configuration timeout interrupt is asserted if the CFG\_TO\_STAT bit of the FCCU\_IRQ\_STAT register is set and the CFG\_TO\_IEN bit of the FCCU\_IRQ\_EN register is also set.

Address: 0h base + E4h offset = E4h

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|----|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14         | 15 |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0  |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30         | 31 |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    | 0  | 0  | CFG_TO_IEN |    |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0          | 0  |

**FCCU\_IRQ\_EN field descriptions**

| Field             | Description                                                                                                                       |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| 0–27<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                           |
| 28<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                           |
| 29–30<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                           |
| 31<br>CFG_TO_IEN  | Configuration Timeout Interrupt Enable<br>0 Configuration timeout interrupt disabled<br>1 Configuration timeout interrupt enabled |

## 55.9.21 X Timer (FCCU\_XTMR)

See [XTMR](#).

This table shows how each timer's state and value vary by FCCU state:

|                           |           | FCCU state                |                                           |                           |                     |
|---------------------------|-----------|---------------------------|-------------------------------------------|---------------------------|---------------------|
| Timer (X)                 | Operation | Configuration             | Normal                                    | Alarm                     | Fault               |
| Alarm-state (ALARM)       | OP17      | Idle (0000_0000h)         | Idle (initial value: <a href="#">TO</a> ) | Running (value when read) | Idle (end of count) |
| Configuration-state (CFG) | OP19      | Running (value when read) | Idle (0001_FFFFh)                         | Idle (0001_FFFFh)         | Idle (0001_FFFFh)   |

## Register descriptions

Address: 0h base + E8h offset = E8h

|       |      |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | XTMR |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |      |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0    | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

### FCCU\_XTMR field descriptions

| Field        | Description                                                                                                                                                                                |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>XTMR | X Timer<br><br>Applies only after successfully running the operation for the associated timer ( <a href="#">OPR</a> ). Specifies the value of the associated timer (X) in CLKSAFE periods. |

## 55.9.22 Transient Configuration Lock (FCCU\_TRANS\_LOCK)

Writable only by code running in Supervisor mode. Temporarily locks and unlocks the configuration. Locking the configuration prevents FCCU from entering Configuration state.

Address: 0h base + F0h offset = F0h

|       |          |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |    |    |    |    |
|-------|----------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|----|----|----|----|----|----|
| Bit   | 0        | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22       | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | Reserved |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |    |    |    |    |    |    |
| W     |          |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    | TRANSKEY |    |    |    |    |    |    |    |    |    |
| Reset | 0        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

### FCCU\_TRANS\_LOCK field descriptions

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                        |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–22<br>Reserved  | This field is reserved.                                                                                                                                                                                                                                                                                                                                                                                            |
| 23–31<br>TRANSKEY | Transient Configuration Lock<br><br>Writable only by code running in Supervisor mode. Temporarily locks and unlocks the configuration. Locking the configuration prevents FCCU from entering Configuration state.<br><br><b>NOTE:</b> You can write to this field when FCCU is in any state, but the lock will not go into effect until FCCU is in Normal state.<br><br>BCh: Unlock.<br><br>Any other value: Lock. |



### 55.9.23 Permanent Configuration Lock (FCCU\_PERMNT\_LOCK)

Writable only by code running in Supervisor mode. Permanently locks the configuration, which prevents FCCU from entering Configuration state until FCCU is reset.

Address: 0h base + F4h offset = F4h

|       |          |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |           |    |    |    |    |    |    |    |    |    |
|-------|----------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|-----------|----|----|----|----|----|----|----|----|----|
| Bit   | 0        | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22        | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | Reserved |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |           |    |    |    |    |    |    |    |    |    |
| W     |          |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    | PERMNTKEY |    |    |    |    |    |    |    |    |    |
| Reset | 0        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0         | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

#### FCCU\_PERMNT\_LOCK field descriptions

| Field              | Description                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–22<br>Reserved   | This field is reserved.                                                                                                                                                                                                                                                                                                                                                                                           |
| 23–31<br>PERMNTKEY | <p>Permanent Configuration Lock</p> <p>Writable only by code running in Supervisor mode. Permanently locks the configuration, which prevents FCCU from entering Configuration state until FCCU is reset.</p> <p><b>NOTE:</b> You can write to this field when FCCU is in any state, but the lock will not go into effect until FCCU is in Normal state.</p> <p>FFh: Lock.</p> <p>Any other value: Do nothing.</p> |

### 55.9.24 Delta T (FCCU\_DELTA\_T)

The FCCU\_DELTA\_T register is used for programming the value of delta\_T constant, in microseconds.

#### NOTE

This register can be written only when the FCCU is in CONFIG state.

#### NOTE

Reserved bits should always be written as all 0's.

Address: 0h base + F8h offset = F8h

|       |          |   |          |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|----------|---|----------|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0        | 1 | 2        | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | Reserved |   | Reserved |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |          |   |          |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0        | 0 | 0        | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

## Register descriptions

|       |          |    |         |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----------|----|---------|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 16       | 17 | 18      | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | Reserved |    | DELTA_T |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |          |    |         |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0        | 0  | 0       | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

### FCCU\_DELTA\_T field descriptions

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–1<br>Reserved   | This field is reserved.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 2–15<br>Reserved  | This field is reserved.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 16–17<br>Reserved | This field is reserved.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 18–31<br>DELTA_T  | <p>Bistable Minimum Fault-Output (EOUT) Faulty Interval</p> <p>Applies only to Bistable fault-output mode (FOM). Controls the minimum amount of time (T<sub>min</sub>) that the fault-output (EOUT) timer runs according to this equation:</p> $T_{min} = 250 \mu s + DELTA\_T$ <p>The following values and meanings are for DELTA_T:</p> <p>00_0000_0000_0000: 0 <math>\mu</math>s</p> <p>00_0000_0000_0001: 1 <math>\mu</math>s</p> <p>...</p> <p>11_1111_1111_1110: 16,382 <math>\mu</math>s</p> <p>11_1111_1111_1111: 16,383 <math>\mu</math>s</p> <p><b>NOTE:</b> The durations shown for the DELTA_T values are for CLKSAFE clock signals running at exactly 16 MHz. However, the CLKSAFE signals are sourced from an internal chip clock signal (for example, IRC)—or an integer division of it (see the chip-specific FCCU information)—that has a trimmed frequency variation. Therefore, the DELTA_T durations may vary. See your chip's data sheet for the trimmed frequency variation (for example, <math>\delta F_{var}</math>).</p> |

## 55.9.25 IRQ Alarm Enable (FCCU\_IRQ\_ALARM\_ENn)

These registers enable the corresponding IRQ alarm.

| IRQ Alarm Enable<br>(FCCU_IRQ_ALARM_ENn) register<br>(value of n) | Offset | IRQENx fields (value of x)      |                                   |
|-------------------------------------------------------------------|--------|---------------------------------|-----------------------------------|
|                                                                   |        | Most significant (leftmost) bit | Least significant (rightmost) bit |
| 0                                                                 | 0FCh   | 31                              | 0                                 |
| 1                                                                 | 100h   | 63                              | 32                                |

**NOTE**

These registers can be written only when the FCCU is in CONFIG state.

Address: 0h base + FCh offset + (4d × i), where i=0d to 1d

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**FCCU\_IRQ\_ALARM\_ENn field descriptions**

| Field          | Description                                                                                                                                                                                                                                           |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>IRQENx | <p>IRQ alarm enable</p> <p>For the mapping of the <a href="#">IRQENx</a> fields among registers, see the earlier table in this register description.</p> <p>0 Alarm is disabled for error source x.</p> <p>1 Alarm is enabled for error source x.</p> |

**55.9.26 NMI Enable (FCCU\_NMI\_ENn)**

These registers enable the NMI.

[Table 55-53](#) shows FCCU NMI enable register channels.

**Table 55-55. FCCU NMI enable register channels**

| Address offset | Register name | Channel range (x)<br>(bit location [0:31]) |
|----------------|---------------|--------------------------------------------|
| 10Ch           | FCCU_NMI_EN0  | NMIENE[31:0]                               |
| 110h           | FCCU_NMI_EN1  | NMIENE[63:32]                              |

**NOTE**

These registers can be written only when the FCCU is in CONFIG state.

Address: 0h base + 10Ch offset + (4d × i), where i=0d to 1d

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**FCCU\_NMI\_ENn field descriptions**

| Field          | Description                                                                                                  |
|----------------|--------------------------------------------------------------------------------------------------------------|
| 0–31<br>NMIENx | <p>NMI enable</p> <p>See <a href="#">Table 55-53</a> for register offset to channel number relationship.</p> |

## FCCU\_NMI\_ENn field descriptions (continued)

| Field | Description                               |
|-------|-------------------------------------------|
| 0     | NMI is disabled for error (NCF) source x. |
| 1     | NMI is enabled for error (NCF) source x.  |

## 55.9.27 Noncritical Fault-State EOUT Signaling Enable (FCCU\_EOUT\_SIG\_ENn)

See [EOUTENx](#).

| Noncritical Fault-State EOUT Signaling Enable (FCCU_EOUT_SIG_ENn) register (value of n) | Offset | EOUTENx fields (value of x)     |                                   |
|-----------------------------------------------------------------------------------------|--------|---------------------------------|-----------------------------------|
|                                                                                         |        | Most significant (leftmost) bit | Least significant (rightmost) bit |
| 0                                                                                       | 11Ch   | 31                              | 0                                 |
| 1                                                                                       | 120h   | 63                              | 32                                |

Address: 0h base + 11Ch offset + (4d × i), where i=0d to 1d

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

## FCCU\_EOUT\_SIG\_ENn field descriptions

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>EOUTENx | <p>Noncritical Fault-State EOUT Signaling Enable x</p> <p>Writable only when FCCU is in Configuration state. Changed by FCCU to its reset value when a Configuration-state timeout occurs. Applies only when the EOUT signals are active (<a href="#">FCCU_SET_AFTER_RESET</a>). When FCCU is configured for Bistable fault-output mode (<a href="#">FOM</a>), controls whether EOUT signaling is enabled as a Fault-state reaction for the associated noncritical fault channel (x). (For other fault-output modes, EOUT signaling is always enabled, regardless of the value of this field.) When EOUT signaling is enabled for an enabled noncritical fault channel, a fault on that channel causes FCCU to indicate the Faulty condition on the EOUT[1:0] signals when FCCU enters Fault state.</p> <p>For all fault-output modes, also controls whether FCCU asserts the FIF signal when a fault on the associated noncritical fault channel (x) causes FCCU to enter Fault state.</p> <p><b>NOTE:</b> For all fault-output modes, you must set this field to enabled to ensure that FCCU asserts the FIF signal when FCCU enters Fault state as the result of a fault on the associated noncritical fault channel (x) so the FOSU module doesn't mistakenly generate a destructive chip reset.</p> <p>For the mapping of the <a href="#">EOUTENx</a> fields among registers, see the earlier table in this register description.</p> <p>0 (in Bistable fault-output mode): EOUT signaling disabled; FIF assertion disabled</p> <p>0 (in other fault-output modes): EOUT signaling enabled; FIF assertion disabled</p> <p>1 (in Bistable fault-output mode): EOUT signaling enabled; FIF assertion enabled</p> <p>1 (in other fault-output modes): EOUT signaling enabled; FIF assertion enabled</p> |

**FCCU\_EOUT\_SIG\_EN $n$  field descriptions (continued)**

| Field | Description |
|-------|-------------|
|-------|-------------|

**55.9.28 Configuration registers****55.9.28.1 Definition**

*Configuration registers* are registers that:

- Let you configure FCCU's Alarm-state timer interval, fault channels, and fault-output (EOUT) signals.
- You can write to only when the configuration is not locked and FCCU is in Configuration state (see [Put FCCU in Configuration or Normal state](#)).
- Save the values you write to them while FCCU is in Configuration state only after you manually put FCCU in Normal state. If FCCU automatically leaves Configuration state and enters Normal state because the configuration-timer interval (TO) expires (called a Configuration-state timeout), FCCU changes the value of the FCCU\_CFG register to its Configuration-state-timeout value and the value of each of the other configuration registers to its reset value. For information on the Configuration-state timeout value, see [FCCU\\_CFG register bit value sources \(N and C\) by event](#).

**55.9.28.2 The configuration registers**

The following registers are the configuration registers, listed in offset order from lowest to highest:

- [Configuration \(FCCU\\_CFG\)](#)
- [Noncritical Fault Configuration \(FCCU\\_NCF\\_CFG \$n\$ \)](#)
- [Noncritical Fault State Configuration \(FCCU\\_NCFS\\_CFG \$n\$ \)](#)
- [Noncritical Fault Enable \(FCCU\\_NCF\\_En\)](#)
- [Noncritical Fault Timeout Enable \(FCCU\\_NCF\\_TOEn\)](#)
- [Noncritical Fault Timeout \(FCCU\\_NCF\\_TO\)](#)
- [Delta T \(FCCU\\_DELTA\\_T\)](#)
- [IRQ Alarm Enable \(FCCU\\_IRQ\\_ALARM\\_EN \$n\$ \)](#)
- [NMI Enable \(FCCU\\_NMI\\_EN \$n\$ \)](#)
- [Noncritical Fault-State EOUT Signaling Enable \(FCCU\\_EOUT\\_SIG\\_EN \$n\$ \)](#)

## 55.9.29 FCCU\_CFG register bit value sources (N and C) by event

Certain events affect the bit values of the [Configuration \(FCCU\\_CFG\)](#) register; when one of these events occurs, the bit value is either unaffected, or FCCU changes it to one of the following:

- N: a chip-specific (or, for some chips, user-specified) value provided by the bit's associated NVMCFG signal
- C: a chip-specific value provided by the bit's associated constant inside the module

For each bit in the [Configuration \(FCCU\\_CFG\)](#) register, this table shows the associated NVMCFG signal, if any, and indicates the source of the bit value for the events that affect it. For specific values of *N* and *C* for each bit, see the chip-specific FCCU information.

| <a href="#">Configuration (FCCU_CFG)</a><br>register bit | NVMCFG signal | Bit value source for<br>an<br>FCCU reset | Bit value source for<br>a<br>configuration-state<br>timeout | Bit value source for<br>an<br>OP31 operation |
|----------------------------------------------------------|---------------|------------------------------------------|-------------------------------------------------------------|----------------------------------------------|
| 0                                                        | 21            | N                                        | C                                                           | N                                            |
| 1                                                        | 20            | N                                        | C                                                           | N                                            |
| 2                                                        | 19            | N                                        | C                                                           | N                                            |
| 3                                                        | 18            | N                                        | C                                                           | N                                            |
| 4                                                        | 17            | N                                        | C                                                           | N                                            |
| 5                                                        | 16            | N                                        | C                                                           | N                                            |
| 6                                                        | 15            | N                                        | C                                                           | N                                            |
| 7                                                        | —             | C                                        | C                                                           | (Unaffected)                                 |
| 8                                                        | —             | C                                        | C                                                           | (Unaffected)                                 |
| 9                                                        | —             | C                                        | C                                                           | (Unaffected)                                 |
| 10                                                       | —             | C                                        | C                                                           | (Unaffected)                                 |
| 11                                                       | —             | C                                        | C                                                           | (Unaffected)                                 |
| 12                                                       | —             | C                                        | C                                                           | (Unaffected)                                 |
| 13                                                       | —             | C                                        | C                                                           | (Unaffected)                                 |
| 14                                                       | —             | C                                        | C                                                           | (Unaffected)                                 |
| 15                                                       | —             | C                                        | C                                                           | (Unaffected)                                 |
| 16                                                       | 14            | N                                        | C                                                           | N                                            |
| 17                                                       | 13            | N                                        | C                                                           | N                                            |
| 18                                                       | 12            | N                                        | C                                                           | N                                            |
| 19                                                       | —             | C                                        | C                                                           | (Unaffected)                                 |
| 20                                                       | 11            | N                                        | C                                                           | N                                            |
| 21                                                       | 10            | N                                        | C                                                           | N                                            |
| 22                                                       | 9             | N                                        | C                                                           | N                                            |
| 23                                                       | 8             | N                                        | C                                                           | N                                            |

*Table continues on the next page...*

| Configuration<br>(FCCU_CFG)<br>register bit | NVMCFG signal | Bit value source for<br>an<br>FCCU reset | Bit value source for<br>a<br>configuration-state<br>timeout | Bit value source for<br>an<br>OP31 operation |
|---------------------------------------------|---------------|------------------------------------------|-------------------------------------------------------------|----------------------------------------------|
| 24                                          | 7             | N                                        | C                                                           | N                                            |
| 25                                          | 6             | N                                        | C                                                           | N                                            |
| 26                                          | 5             | N                                        | C                                                           | N                                            |
| 27                                          | 4             | N                                        | C                                                           | N                                            |
| 28                                          | 3             | N                                        | C                                                           | N                                            |
| 29                                          | 2             | N                                        | C                                                           | N                                            |
| 30                                          | 1             | N                                        | C                                                           | N                                            |
| 31                                          | 0             | N                                        | C                                                           | N                                            |

## 55.10 FCCU Output Supervision Unit

The FOSU provides a supervision of the primary fault notification path by analyzing FCCU behavior for correctness. It waits for any reaction of the FCCU in a fixed time window after a fault is signaled.

The intention of the FOSU is to provide a secondary fault reaction path in most cases when the FCCU fails but not to needlessly propagate a fault which is already handled by the FCCU in a full chip reset. Only a failed primary fault reaction (that is, FCCU's failure) is a reason for the secondary reaction to take over (and generate a reset request).

There is a 'do nothing' input coming from the FCCU that indicates that the FCCU is programmed for no reaction for ALL FAULTS. It is a "static" input in the sense that it does not change after FCCU configuration. The FOSU masks the incoming faults with the 'do nothing' control from the FCCU, meaning that a fault is not captured by the FOSU if the 'do nothing' signal is asserted, (i.e., a disabled fault). There is no minimum pulse width requirement on the fault indication other than what is required by the technology, which is the same as that of the FCCU. FOSU does not monitor FCCU for the case of faults occurring during CONFIG state.

The FOSU contains a timer with a duration of FOSU\_COUNT, driven by CLKSAFE. The timer is initialized and started on any captured, enabled fault. While the timer is running, any subsequent captured fault will neither restart nor reinitialize the timer. The timer is stopped when the FCCU shows any of the following reactions (the FOSU does not check whether the reaction is the configured one for the faults which occurred):

- Functional reset
- IRQ (triggered by ALARM state)

- NMI
- Error out triggered (by FCCU or by SW)

When the timer is stopped, the fault capture logic is cleared in order to ensure that the timer is not restarted due to faults still 'stuck' in the capture logic. The timer will then be restarted by the next new failure indication. When the timer expires, the FOSU's failure indicator output is asserted after it ensures that the fault is enabled and the static "fccu program to do nothing" signal is deasserted. This is because FCCU uses settings after it exits CONFIG state, even if fault captured before the exit.

The FOSU's failure indication output is connected to a reset input of the system reset state machine and is treated like a power-on reset event. When this reset is asserted, the FOSU's capture logic is cleared, its timer is kept stopped and in a non-expired state, and its failure indicator output is deasserted.

### NOTE

FOSU is triggered on assertion of enabled fault. In case the triggering fault is disabled, FOSU times-out without reaction.

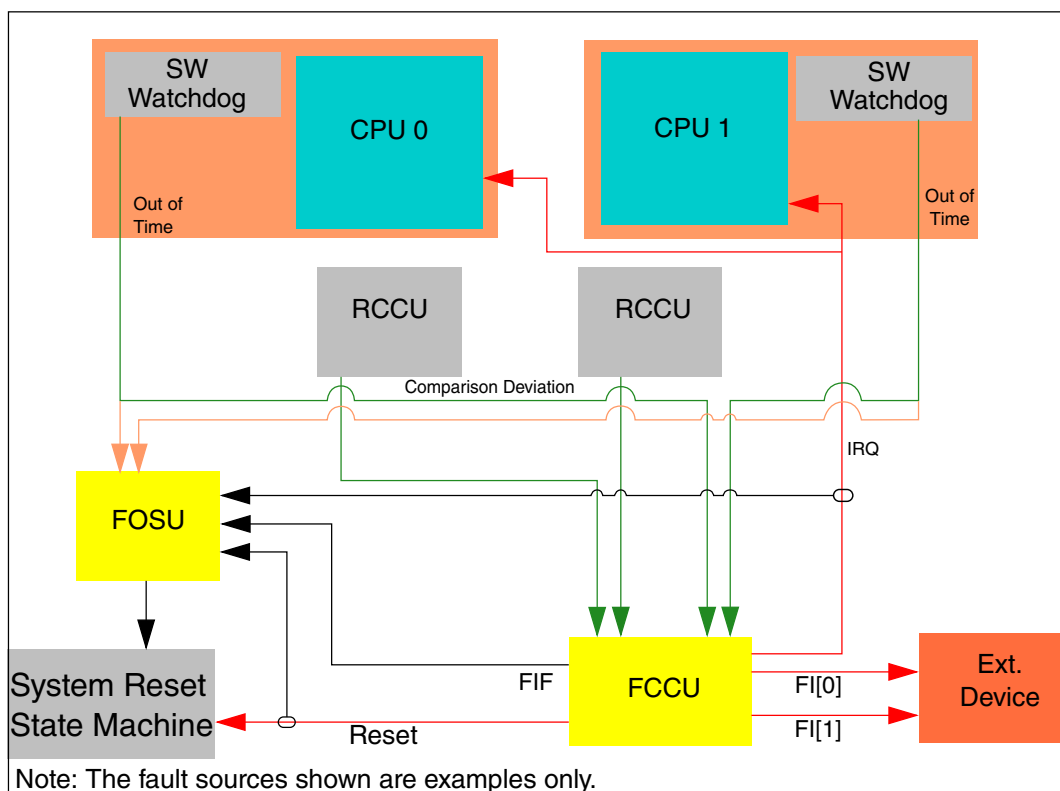


**Figure 55-56. FOSU connections to the FCCU and system reset state machine**



It is important to remember that there will be no FCCU reaction to a fault while the FCCU is in the CONFIG state. For this reason, the FCCU should not be kept in CONFIG for longer than the FOSU\_COUNT duration. Otherwise, there is a risk that an incoming error report causing the FOSU to mistakenly see the FCCU as having failed, and then resets the MCU.

### NOTE

FCCU and FOSU timeout counter settings: The FCCU counter value should always be programmed less than the FOSU counter value, FOSU\_COUNT, or resets may be generated by an FOSU time-out. This ensures that the FOSU reacts only when the FCCU fails to react to any particular fault.

## 55.11 Use cases and limitations

### 55.11.1 Configuration guidelines

Follow these guidelines to configure FCCU:

- If you want FCCU to react to a fault on a noncritical fault channel:
  - Enable the channel ([NCFEx](#)).
  - Enable at least one type of Fault-state reaction for the channel: chip reset ([NCFSCx](#)), fault-output (EOUT) signaling ([EOUTENx](#)), or nonmaskable interrupt ([NMIENx](#)).
  - If you enable chip reset as the type of Fault-state reaction for the channel ([NCFSCx](#)), enable either Alarm state ([NCFTOEx](#)) or at least one other type of Fault-state reaction for the channel: fault-output (EOUT) signaling ([EOUTENx](#)) or nonmaskable interrupt ([NMIENx](#)).
  - If you enable Alarm state for the channel ([NCFTOEx](#)), enable the Alarm-state reaction ([IRQENx](#)).
  - If you enable Alarm state for the channel ([NCFTOEx](#)), make sure the Alarm-state timer interval ([TO](#)) is less than the FOSU module's timer interval; otherwise, FOSU generates a chip reset every time a fault occurs on the channel. The FOSU timer interval (FOSU\_COUNT) is chip-specific. See the chip-specific FCCU information.

### NOTE

If you enable a noncritical fault channel but disable all reactions for that channel, FCCU changes state if necessary

but doesn't perform any reaction because reactions are disabled. If you enable reactions for a noncritical fault channel but disable that channel, and FCCU is in Normal state when a fault occurs on the channel, FCCU doesn't enter Alarm or Fault state and therefore doesn't perform any reaction.

### 55.11.2 Recommendations to configure FCCU

1. After a reset cycle in which the FOSU, FCCU, and system are initialized (such as caused by power-up, RESET pin, FOSU reset, or STCU reset), the following steps could be followed to configure FCCU:
  - a. Check and clear any pending fault status
  - b. Verify FCCU is in NORMAL state, else repeat step(a) above
  - c. Configure FCCU
2. After a reset cycle in which the system is re-initialized but the FOSU and RCCU are not re-initialized, the following steps could be followed to reconfigure FCCU:
  - a. If active, wait for the Error out T\_min to expire
  - b. Check and clear fault status
  - c. Error pin moves to "non faulty" state, once fault status is cleared and T\_min expires
  - d. Verify FCCU is in NORMAL state, else repeat step(a) above
  - e. Read and verify value in FCCU\_NCF\_Ex
  - f. Reconfigure FCCU, if necessary

## Chapter 56

# Self-Test Control Unit (STCU2)

### 56.1 Chip-specific STCU information

The STCU controls the execution of built-in self-tests (BISTs) of the chip.

The STCU manages two primary types of BISTs:

- MBIST: memory (SRAM/ROM) BIST
- LBIST: logic BIST for digital logic

The STCU has two sets of conditions under which it applies a self-test sequence:

- Off-line condition: After the user stores self-test parameters in flash memory and a reset cycle is initiated by a power-up event, RESET pin assertion, or FOSU reset, the STCU loads the stored parameters, executes a self-test, and then resets the chip.
- On-line condition: After the user initiates one or more self-tests during normal chip operation, the STCU executes the self-tests and then resets the chip.

The STCU controls the clocking and the reset of the chip during a self-test.

#### NOTE

If both MBIST and LBIST are scheduled to be executed, then MBIST should be executed first. LBIST can be executed only after MBIST is completed.

#### 56.1.1 Supported BIST sequences

NXP has validated and therefore supports only particular STCU2 built-in self-test (BIST) sequences for this chip. For descriptions of the supported BIST sequences for this chip, see the Application Note titled *MPC5777C STCU Quick Start Guide* (document AN5288).

This chip does not have a default offline BIST sequence.

### 56.1.2 Configure or bypass STCU2 off-line self-tests

During reset, the STCU2 has a watchdog timer that can time out and delay exit from reset by at least ~8 ms. To prevent this delay, program DCF records either to configure and run or to bypass STCU2 off-line self-tests at startup.

After programming one set of DCF records either to run or bypass off-line self-tests, do not program subsequent DCF records to change the initial selection of run or bypass.

For information about configuring and running self-tests, see [Supported BIST sequences](#).

To bypass STCU2 off-line self-test operation:

1. Program a DCF record for the client STCU\_SKC (see [Table 5-6](#)) to write the Key1 and Key2 codes for off-line self-test. See [STCU2 SK Code Register \(STCU2\\_SKC\)](#) for details about the register.
2. Program a DCF record for the client STCU\_RUN (see [Table 5-6](#)) to write 1 to STCU2\_RUN[BYP]. See [STCU2 Run Register \(STCU2\\_RUN\)](#) for details about STCU2\_RUN[BYP].

### 56.1.3 STCU interfaces to other modules

The STCU interfaces with other modules on the chip, including:

- SSCM: For off-line self-tests, the STCU uses the SSCM DCF bus to load self-test parameters from DCF records in flash memory. After the reset that precedes the off-line self-test, this SSCM interface is able to write the configuration parameters and start the off-line self-test execution only one time.
- IPS bus (Peripheral Bridge):
  - For on-line self-tests, the CPU accesses STCU memory-mapped registers to initiate execution of the self-tests.
  - For both off-line and on-line self-tests, the CPU accesses STCU memory-mapped registers to check self-test results.
- FCCU: For off-line self-tests, the STCU can be configured through DCF records to classify a hardware fault as a critical fault (CF) or non-critical fault (NCF).

- If a subsequent off-line self-test detects a hardware fault that is classified as a NCF, the FCCU's default response is no reaction.
- If a subsequent off-line self-test detects a hardware fault that is classified as a CF, the FCCU responds to the CF by initiating a new reset cycle.

### NOTE

On this chip, the critical fault (CF) output of the STCU is connected to a non-critical fault (NCF) input of the FCCU. This input is programmed to issue a reset by default. The FCCU itself is not configured to receive "native" CF inputs.

## 56.1.4 On-line LBIST and MBIST clock configuration

Before running an on-line LBIST or MBIST, configure clocking as follows to ensure proper execution:

1. PLL0 is the global clock source: Set PLL0 dividers to generate 200 MHz at PLL0's output.
2. Configure PLL0's clock source to be the crystal oscillator (XOSC): Set SIU\_SYSDIV[PLL0SEL] to 0.
3. Select the non-FM PLL, PLL0, to drive the system clock: set SIU\_SYSDIV[SYSCLKSEL] to 11b.
4. Disable the Progressive Clock Switch feature: Set SIU\_SYSDIV[PCSEN] to 0.
5. Set the System Clock Divider to divide by 1: Set SIU\_SYSDIV[SYSCLKDIV] to 1xx.
6. Set the Peripheral Clock Select: Set SIU\_SYSDIV[PERCLKSEL] to 0.
7. Ensure the fm\_per\_clk divider is 2: Ensure SIU\_SYSDIV[FMPERDIV] is 00b.
8. Set the LFAST clock source to be the peripheral clock coming from PLL: Set SIU\_LFCLKCFG[LFCLKSEL] to 10b.
9. Set SIU\_SYSDIV[MCANSEL] to 1.
10. Set FlexCAN's CAN\_CTRL1[CLKSRC] to 1.
11. Set SWT\_A's SWT\_CR[CSL] to 0 and SWT\_B's SWT\_CR[CSL] to 0.

In addition, before executing an on-line MBIST, set the following clock divider values:

1. Set the eTPU clock divider to 2: Set SIU\_SYSDIV[ETPUDIV] to 0.
2. Set the TCK divider to 3: Set STCU2\_CFG[CLK\_CFG] to 010b.

In addition, before executing an on-line LBIST:

- Configure the shift speed clock for the applicable LBIST with a divider equal to or greater than 2: Set the applicable STCU2\_LB\_CTRL $n$ [SHS] field to a value of 001b or greater.

During an on-line LBIST, clock dividers are bypassed by default.

### 56.1.5 STCU2\_PLL\_CFG configuration of PLLDIG\_PLL0DV

When used, the STCU2's [PLL interface](#) enables the STCU2 module to control PLL0 during off-line self-test operations. In this case, the values programmed into the fields of the [STCU2 PLL Configuration Register \(STCU2\\_PLL\\_CFG\)](#) configure the corresponding fields of the [PLLDIG PLL0 Divider Register \(PLLDIG\\_PLL0DV\)](#).

The following table shows the correlation between STCU2\_PLL\_CFG fields and PLLDIG\_PLL0DV fields.

**Table 56-1. STCU2\_PLL\_CFG and PLLDIG\_PLL0DV field alignment**

| STCU2_PLL_CFG field    | PLLDIG_PLL0DV field    |
|------------------------|------------------------|
| <a href="#">PLLLDF</a> | <a href="#">MFD</a>    |
| <a href="#">PLLIDF</a> | <a href="#">PREDIV</a> |
| <a href="#">PLLODF</a> | <a href="#">RFDPHI</a> |

### 56.1.6 Wait time for writing to the on-line registers

On this chip, no additional wait time applies to writing the on-line registers.

### 56.1.7 On-line MBIST

After the STCU2 triggers the execution of an on-line MBIST, the user cannot perform any access until the MBIST ends.

At the end of the execution of an on-line MBIST, the STCU2 does not generate any reset automatically. The reason is that the software application must manage the consequences of write operations that the MBIST's execution performs.

After the execution of an on-line MBIST ends, the user must ensure that memory controllers are in the correct state.

- Reset the chip to clear the memory output.
- If any memory content is destroyed by an MBIST, reload the content after the MBIST.

It is recommended to enable MBIST interrupts only when executing MBISTs exclusively, not MBISTs combined with LBISTs. If an MBIST is combined with an LBIST in the same test run, the LBIST triggers a reset after the test execution, and the MBIST interrupt will be pending coming out of reset.

This chip implements 65 MBIST partitions.

**Table 56-2. MBIST partitions**

| MBIST partition | Memory                   |
|-----------------|--------------------------|
| MBIST0          | SRAM 1_4                 |
| MBIST1          | SRAM 1_3                 |
| MBIST2          | SRAM 1_2                 |
| MBIST3          | SRAM 1_1                 |
| MBIST4          | SRAM 2_4                 |
| MBIST5          | SRAM 2_3                 |
| MBIST6          | SRAM 2_2                 |
| MBIST7          | SRAM 2_1                 |
| MBIST8          | FEC FIFO RAM             |
| MBIST9          | FEC MIB RAM              |
| MBIST10         | DMA RAM 2                |
| MBIST11         | DMA RAM 1                |
| MBIST12         | eTPU N3 SCM RAM 2        |
| MBIST13         | eTPU N3 SCM RAM 1        |
| MBIST14         | eTPU S32D SCM RAM        |
| MBIST15         | eTPU N3 SPM RAM 3        |
| MBIST16         | eTPU N3 SPM RAM 2        |
| MBIST17         | eTPU N3 SPM RAM 1        |
| MBIST18         | eTPU S32D SPM RAM 2      |
| MBIST19         | eTPU S32D SPM RAM 1      |
| MBIST20         | CSE RAM                  |
| MBIST21         | Core1 cache data RAM 1_4 |

*Table continues on the next page...*

**Table 56-2. MBIST partitions (continued)**

| <b>MBIST partition</b> | <b>Memory</b>            |
|------------------------|--------------------------|
| MBIST22                | Core1 cache data RAM 1_3 |
| MBIST23                | Core1 cache data RAM 1_2 |
| MBIST24                | Core1 cache data RAM 1_1 |
| MBIST25                | Core1 cache data RAM 2_4 |
| MBIST26                | Core1 cache data RAM 2_3 |
| MBIST27                | Core1 cache data RAM 2_2 |
| MBIST28                | Core1 cache data RAM 2_1 |
| MBIST29                | Core0 cache data RAM 1_4 |
| MBIST30                | Core0 cache data RAM 1_3 |
| MBIST31                | Core0 cache data RAM 1_2 |
| MBIST32                | Core0 cache data RAM 1_1 |
| MBIST33                | Core0 cache data RAM 2_4 |
| MBIST34                | Core0 cache data RAM 2_3 |
| MBIST35                | Core0 cache data RAM 2_2 |
| MBIST36                | Core0 cache data RAM 2_1 |
| MBIST37                | Core1 cache tag RAM 1_4  |
| MBIST38                | Core1 cache tag RAM 1_3  |
| MBIST39                | Core1 cache tag RAM 1_2  |
| MBIST40                | Core1 cache tag RAM 1_1  |
| MBIST41                | Core1 cache tag RAM 2_4  |
| MBIST42                | Core1 cache tag RAM 2_3  |
| MBIST43                | Core1 cache tag RAM 2_2  |
| MBIST44                | Core1 cache tag RAM 2_1  |
| MBIST45                | Core0 cache tag RAM 1_4  |
| MBIST46                | Core0 cache tag RAM 1_3  |
| MBIST47                | Core0 cache tag RAM 1_2  |
| MBIST48                | Core0 cache tag RAM 1_1  |
| MBIST49                | Core0 cache tag RAM 2_4  |
| MBIST50                | Core0 cache tag RAM 2_3  |
| MBIST51                | Core0 cache tag RAM 2_2  |
| MBIST52                | Core0 cache tag RAM 2_1  |
| MBIST53                | Core1 cache "dirty" RAM  |
| MBIST54                | Core0 cache "dirty" RAM  |
| MBIST55                | FlexCAN RAM 1_2          |
| MBIST56                | FlexCAN RAM 1_1          |
| MBIST57                | FlexCAN RAM 2_2          |
| MBIST58                | FlexCAN RAM 2_1          |
| MBIST59                | M_CAN RAM                |
| MBIST60                | eTPU Nexus RAM 1_2       |

*Table continues on the next page...*



**Table 56-2. MBIST partitions (continued)**

| MBIST partition | Memory             |
|-----------------|--------------------|
| MBIST61         | eTPU Nexus RAM 1_1 |
| MBIST62         | eTPU Nexus RAM 2_2 |
| MBIST63         | eTPU Nexus RAM 2_1 |
| MBIST64         | BAM ROM            |

## 56.1.8 LBIST scenarios

This chip implements six logical partitions, and each one has an LBIST Controller. The STCU controls the LBIST execution of all partitions.

**Table 56-3. LBIST partition mapping**

| LBIST partition | Modules                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| LBIST0          | Core1                                                                                                                                                                                                                                                                                                                                                                                                          |
| LBIST1          | Core0                                                                                                                                                                                                                                                                                                                                                                                                          |
| LBIST2          | Checker core, RCCU                                                                                                                                                                                                                                                                                                                                                                                             |
| LBIST3          | Peripheral Bridge B (AIPS_0) and associated peripherals: <ul style="list-style-type: none"> <li>• SDADC_1 and SDADC_3</li> <li>• M_CAN Subsystem</li> <li>• CRC</li> <li>• DSPI_A, DSPI_B, and DSPI_C</li> <li>• eMIOS_0</li> <li>• EQADC_A</li> <li>• eSCI_A and eSCI_B</li> <li>• eTPU_C</li> <li>• FCCU including FOSU</li> <li>• FlexCAN_0 and FlexCAN_1</li> <li>• IGF_0</li> <li>• SRX_0</li> </ul>      |
| LBIST4          | Peripheral Bridge A (AIPS_1) and associated peripherals: <ul style="list-style-type: none"> <li>• SDADC_2 and SDADC_4</li> <li>• CMU_0 to CMU_8</li> <li>• DSPI_D and DSPI_E</li> <li>• eMIOS_1</li> <li>• EQADC_B</li> <li>• eSCI_D, eSCI_E, and eSCI_F</li> <li>• eTPU_A/B</li> <li>• FCCU including FOSU</li> <li>• FlexCAN_2 and FlexCAN_3</li> <li>• IGF_1</li> <li>• PIT_RTI</li> <li>• SRX_1</li> </ul> |
| LBIST5          | Core platform modules: <ul style="list-style-type: none"> <li>• XBAR</li> <li>• EIM</li> <li>• ERM</li> </ul>                                                                                                                                                                                                                                                                                                  |

**Table 56-3. LBIST partition mapping**

| LBIST partition | Modules                                                                                                                                                                                                                                                                                                                   |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                 | <ul style="list-style-type: none"> <li>• PCU</li> <li>• MPU_0</li> <li>• MPU_1</li> <li>• CSE</li> <li>• FEC</li> <li>• PFLASH</li> <li>• INTC</li> <li>• PRAMC_0</li> <li>• PRAMC_1</li> <li>• eDMA_A</li> <li>• eDMA_B</li> <li>• STM</li> <li>• SWT_A</li> <li>• SWT_B</li> <li>• C55FMC flash memory array</li> </ul> |

### 56.1.8.1 Off-line LBIST

The chip supports off-line LBIST operation for all six LBIST partitions.

Summary of an off-line LBIST sequence:

1. The user stores off-line self-test parameters in a DCF record in flash memory. The self-test can include multiple LBIST partitions.
  - For off-line self-tests, LBIST execution on individual LBIST partitions occurs serially. The chip does not support LBIST execution on individual LBIST partitions in parallel.
  - An off-line LBIST can run with either the 16 MHz IRC or with PLL0.
  - For an off-line self-test, the STCU can enable and program PLL0 using STCU DCFs.
2. The user resets the chip by cycling power or asserting the RESET pin.
3. The STCU loads the stored off-line LBIST parameters from the DCF record via the SSCM interface.
4. The off-line LBIST executes and ends.
5. Hardware issues a functional reset. The STCU is not affected by this reset.
6. The user verifies the FCCU channels corresponding to STCU faults. If any fault is flagged, the detailed cause can be checked by reading STCU status registers.

### 56.1.8.2 On-line LBIST

The chip supports on-line LBIST operation on all six LBIST partitions.

#### NOTE

When LBIST execution begins, the embedded flash memory must be in an idle state; program or erase operations must not be active. See the [Functional Description](#) of the embedded flash memory for details about its states.

Summary of an on-line LBIST sequence:

1. The user initiates an on-line LBIST by programming STCU memory-mapped registers.
  - An on-line LBIST is able to test only one partition at a time.
  - An on-line LBIST can run with the functional operating frequencies. The maximum-frequency clock for the LBIST partition is derived from PLL0.
  - The user can initiate a sequence of more than one on-line LBIST. Multiple on-line LBISTs can execute serially, with each self-test run on an individual partition.
  - Program the LBRMSW register to generate a global functional reset after an on-line LBIST. Avoid using a dedicated reset due to random data injected by the LBIST Controller.
2. Hardware forces all LBIST partitions into a quiescent state.
3. The on-line LBIST executes and ends.
4. After the completion of the LBIST on-line self-test sequence, hardware issues a functional reset. The STCU is not affected by this reset.
5. The user verifies the FCCU channels corresponding to STCU faults. If any fault is flagged, the detailed cause can be checked by reading STCU status registers.

For on-line LBISTs, this chip does not support the following features:

- Interrupt generation after each LBIST on an individual partition
- Software abort during an on-line LBIST

#### NOTE

An abort does occur under these conditions:

- A reset cycle is triggered in the middle of a self-test execution.
- The reset source is not POR, RESET pin, FOSU, or the STCU itself.

A reset caused by POR, RESET pin, or FOSU does not cause an abort because it resets the STCU and causes it to run the self-test again. A reset caused by the STCU itself implies successful completion of the self-test procedure.

### **56.1.8.3 Diagnostic self-test: On-line LBIST from JTAG/Nexus interface**

A "diagnostic self-test" is an on-line LBIST that is invoked via the chip's JTAG/Nexus debug/development interface. Via the chip's JTAG pins, a JTAG tool can access the Nexus debug interface in the CPU and use it to configure the STCU without interfering with normal chip operation.

However, after the LBIST of a partition starts, the STCU takes control of the chip, putting all partitions that are not being tested in a quiescent state. The chip resets at the end of the LBIST.

### **56.1.8.4 Multiple serial LBISTs**

The chip allows serial execution of off-line and on-line LBISTs. For example, this sequence is valid:

1. The user programs a DCF record in the flash memory to perform an off-line LBIST.
2. A power-on reset (POR) occurs.
3. The STCU executes the off-line self-test.
4. A functional reset occurs.
5. The CPU boots. The user programs STCU registers using the CPU to execute an on-line LBIST. An on-line LBIST is initiated.
6. A functional reset occurs.
7. The CPU boots again. No off-line or on-line self-test is initiated.

### 56.1.8.5 LBIST execution and dual cores

On this dual-core chip, either core can program (schedule) execution of an LBIST. After the LBIST of a partition begins, both cores and all other logic not included in that partition are maintained in a reset state. Neither core can operate during the LBIST execution of any partition, regardless of which core initially configured the LBIST.

### 56.1.9 STCU register default values

The following table documents STCU register default values that are specific to this chip. For the default values of all other STCU registers, see [Register description](#).

#### NOTE

Some of these registers are DCF clients whose values are overwritten by the values in STCU2-specific DCF records in UTEST flash memory. If the user programs a value for one of these DCF records, the corresponding register reflects the user-programmed value.

**Table 56-4. STCU register default values**

| Register                                        | Default value |
|-------------------------------------------------|---------------|
| STCU2_CFG                                       | 0000_0000h    |
| STCU2_PLL_CFG                                   | 0000_0000h    |
| STCU2_WDG                                       | 0000_FFFFh    |
| STCU2_ERR_FM                                    | 0000_0000h    |
| STCU2_LBRMSW                                    | 0000_003Fh    |
| STCU2_LBUFM                                     | 0000_0000h    |
| STCU2_MBUFML, STCU2_MBUFMM, STCU2_MBUFMH        | 0000_0000h    |
| STCU2_LB_CTRL $_n$                              | 0000_0000h    |
| STCU2_LB_PCS $_n$                               | 0000_0000h    |
| STCU2_LB_MISREL $_n$ , STCU2_LB_MISREH $_n$     | FFFF_FFFFh    |
| STCU2_LB_MISRRL $_n$ , STCU2_LB_MISRRH $_n$     | 0000_0000h    |
| STCU2_LB_MISRELSW $_n$ , STCU2_LB_MISREHSW $_n$ | FFFF_FFFFh    |
| STCU2_LB_MISRRLSW $_n$ , STCU2_LB_MISRRHSW $_n$ | 0000_0000h    |
| STCU2_MB_CTRL $_n$                              | 0000_0000h    |

## 56.1.10 AUTOLOCK\_VALUE for register write access and DCF\_COMPLETION value for DCF write completion

Register write-access watchdog timer refers to two clock-cycle values related to the STCU2's write-access watchdog. The following table provides these values on this chip.

| Clock-cycle value | Number of STCU2 clock cycles |
|-------------------|------------------------------|
| AUTOLOCK_VALUE    | 1024                         |
| DCF_COMPLETION    | 16                           |

## 56.2 Introduction

The *STCU2 self-test control unit* is a hardware module that:

- Manages the execution of built-in self-tests (BISTs)
- Indicates whether each BIST passed

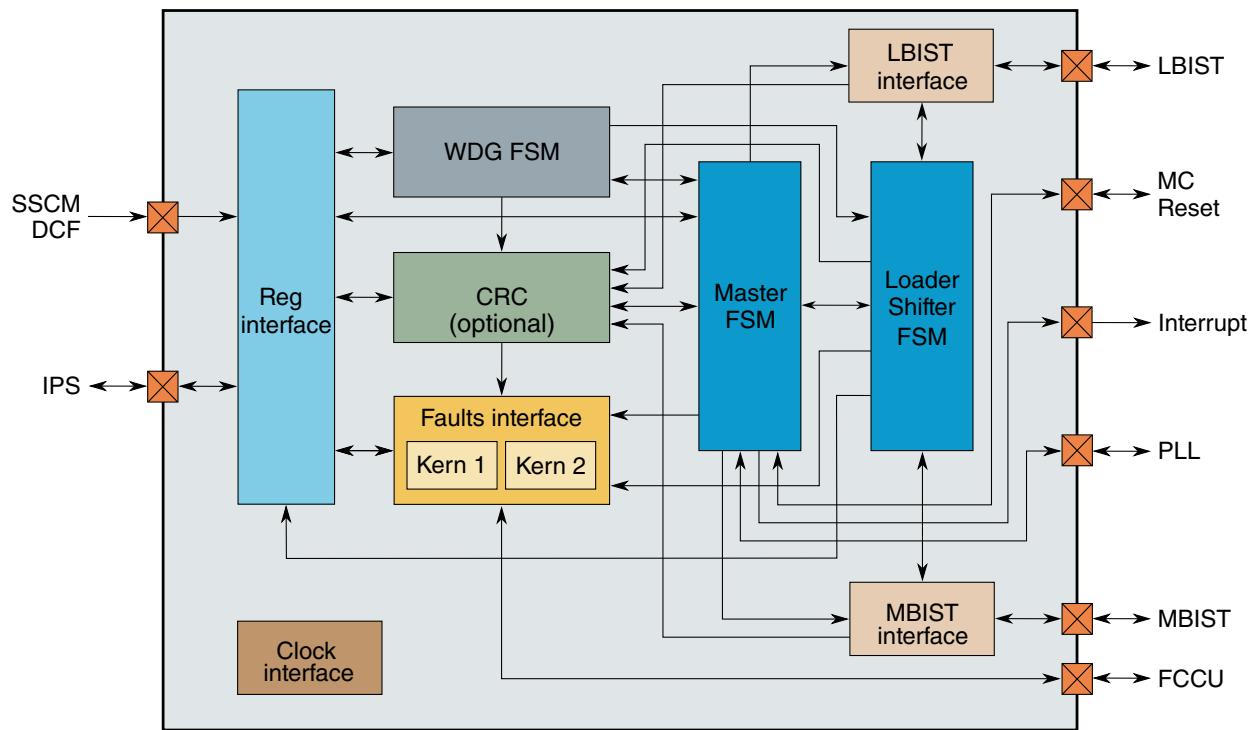
## 56.3 Main features

- Internal Peripheral System (IPS) interface for reading from and writing to the STCU2 registers during runtime (online) via the CPU
- SSCM Device Configuration Format (DCF) interface for writing to the STCU2 registers from nonvolatile memory during reset (offline) via the SSCM module (has priority over the IPS interface)
- Programmable scheduler for BIST execution
- LBIST concurrent/sequential execution
- MBIST concurrent/sequential execution
- Programmable LBIST delayed concurrent start
- Programmable internal clock prescaler to reduce internal and BIST TCK clocks
- Programmable PLL direct control during offline BIST sequence
- PLL Lock signal monitoring during BIST sequences
- Dedicated online MBIST interrupt source generation
- Programmable BIST-sequence-execution watchdog timer that specifies the maximum time allowed for the execution of a BIST sequence

- Fixed offline-initialization watchdog timer that specifies the maximum time allowed for offline initialization before indicating an initialization fault
- Fixed register-write-access watchdog timer that specifies the maximum amount of time that STCU2 allows you to write to its registers after you unlock them
- Fields that indicate the execution status of each:
  - Online MBIST
  - Offline MBIST
  - Online LBIST
  - Offline LBIST
- Fields that indicate the status of each type of online STCU2 internal error condition
- Fields that indicate the status of each type of offline STCU2 internal error condition
- Programmable fault mapping for each BIST for controlling the type of fault that STCU2 reports (recoverable or unrecoverable) when the BIST fails to execute
- Programmable fault mapping for each STCU2 internal error condition for controlling the type of fault that STCU2 reports (recoverable or unrecoverable) when the condition occurs
- Signals for reporting recoverable and unrecoverable faults to the FCCU module
- Redundant recoverable and unrecoverable fault-generation logic to improve reliability
- FCCU recoverable and unrecoverable fault-injection mechanism
- LBIST MISRE register access to perform redundant check by software
- Offline BIST-sequence bypass capability
- Global register write-protection mechanism that requires two security key codes
- Global automatic power saving when a BIST sequence is completed, Bypass mode is selected, or a watchdog timer time-out is detected
- ITF/watchdog automatic clock wake-up mechanism when software unlocks the STCU2 registers for write access
- ITF/watchdog automatic power saving when the fixed register-write-access watchdog timer times out

# 56.4 Block diagram

This diagram shows the parts of STCU2:



This table describes the parts of STCU2:

| Part            | Function                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Reg interface   | Provides access to: <ul style="list-style-type: none"><li>Registers</li><li>The security key logic</li><li>The Internal Peripheral System (IPS) interface</li><li>The SSCM Device Configuration Format (DCF) interface</li></ul>                                                                                                                                                                                                                                                                                                                                                      |
| Fault interface | Does the following: <ul style="list-style-type: none"><li>Collects the fault conditions caused by STCU2 internal error conditions and each BIST executed in a sequence.</li><li>Depending on the fault mapping for a given BIST, sets the global recoverable or unrecoverable status flag.</li><li>Manages the recoverable and unrecoverable fault lines to and from the FCCU module</li><li>Manages the set/clear injection mechanism provided by the FCCU module.</li></ul> <p>To improve the intrinsic reliability of this critical logic, the generation logic is duplicated.</p> |
| Clock interface | Manages the internal and the BIST TCK clock prescaler, the internal clock-gating power saving, and the wake-up clock feature.                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

Table continues on the next page...



| Part                                      | Function                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| WDG FSM (watchdog finite state machine)   | Provides the following: <ul style="list-style-type: none"> <li>• A programmable BIST-sequence-execution watchdog timer that specifies the maximum time allowed for the execution of a BIST sequence</li> <li>• A fixed offline-initialization watchdog timer that specifies the maximum time allowed for offline initialization before indicating an initialization fault</li> <li>• A fixed register-write-access watchdog timer that specifies the maximum amount of time that STCU2 allows you to write to its registers after you unlock them</li> </ul> |
| Master FSM (finite state machine)         | Coordinates and schedules all of the operations performed during a BIST sequence.                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Loader Shifter FSM (finite state machine) | Programs the BIST registers and reads back the data to be checked at the end of each test operation.                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| LBIST interface                           | Provides the interface between the chip's LBIST engines and the STCU2 controllers.                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| MBIST interface                           | Provides the interface between the MBIST controller and the STCU2 controllers.                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |

## 56.5 IPS bus interface

The IPS bus interface is a slave bus used for configuration purposes via CPU. The following bus read operations (contiguous byte enables) are supported:

- Word (32 bits) data read operations to any registers
- Low and high half-words (16 bits, data[31:16] or data[15:0]) data read operations to any registers
- Byte(8 bits, data[31:24] or data[23:16] or data[15:8] or data[7:0]) data read operations to any registers
- Any other operation (free byte enables or other operations) has to be avoided.

The same operations are also supported in write mode only in the following scenario:

- The Self-Test sequence parameters are loaded before Off-Line Self-test is executed
- The IPS access on a specific register is allowed by software only when the bit STCU2\_CFG.WRP is cleared
- In all the other conditions the write operations are not supported and all the STCU2 registers can be only read. Only exceptions are the following read/write bits: STCU2\_CFG.WRP, STCU2\_ERR\_STAT.UFSF and STCU2\_ERR\_STAT.RFSF. See related register description for additional details

The STCU2 module generates a transfer error in the following cases:

- Any write/read access to the register addresses not mapped on the peripheral but included in the address space of the peripheral
- Any write/read operation different from byte/halfword/word (free byte enables or other operations) on each register
- Any write operation on Double Security Key register applying a wrong sequence of keys (the two write operations cannot be interleaved with other access to STCU2 registers)
- Any write operation performed on a register when the Double Security keys have not been applied
- Any write operation performed on registers when the STCU2\_CFG.WRP bit is set and the access is performed through software (IPS interface)
- Any write operation performed through software (IPS interface) on Read/On-Line\_Write/Off-Line registers
- Any write operation performed off-line on registers in which writing operation is allowed only in the On-Line Self-Test phase when STCU2\_CFG[WRP] = 1
- Any write operation performed on Read Only registers

The registers of the STCU2 module are accessible (read/write) in each access mode: user, supervisor or test.

In case there are write operations on bits marked as reserved, the transfer error is not generated.

### **NOTE**

See the chip-specific STCU2 information for the wait time necessary to write to the online registers.

## **56.6 BISTs and BIST partitions**

### **56.6.1 Definition: BIST**

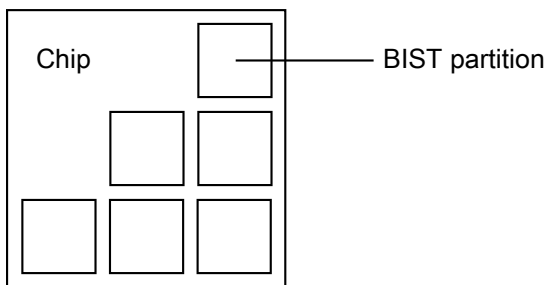
A *built-in self-test (BIST)* is a test that the chip can execute to verify the functional integrity of a part of itself. The chip uses STCU2 and other on-chip hardware to execute a BIST.

## 56.6.2 Definition: BIST partition

A *built-in self-test (BIST) partition* is a part of a chip for which a BIST has been defined. The hardware that is included in a given BIST partition is chip-specific. For a list of the hardware included in each of the BIST partitions on this chip, see the chip-specific STCU2 information.

## 56.6.3 Example: BIST partitions on a chip

This is an example of a chip with six built-in self-test (BIST) partitions:



## 56.6.4 Types of BIST partitions

The STCU2 module on this chip supports the following type or types of built-in self-test (BIST) partitions:

| BIST partition type           | Description                 |
|-------------------------------|-----------------------------|
| Memory BIST (MBIST) partition | An SRAM or ROM block        |
| Logic BIST (LBIST) partition  | One or more digital modules |

## 56.7 BIST sequences

### 56.7.1 Definition: BIST sequence

A *built-in self-test (BIST) sequence* is a programmable series of one or more phases, each of which executes one or more individual BISTs.

## 56.7.2 STCU2 executes MBISTs before LBISTs

If a built-in self-test (BIST) sequence includes both MBISTs and LBISTs, STCU2 executes the MBISTs first.

## 56.7.3 Example: Single-phase BIST sequence

This is an example of a single-phase BIST sequence in which STCU2 executes only one BIST:

| Phase | BIST executed |
|-------|---------------|
| 0     | MBIST 16      |

## 56.7.4 Example: Multiphase BIST sequence

This is an example of a multiphase BIST sequence in which STCU2 executes more than one BIST in parallel in some phases:

| Phase | BISTs executed       |
|-------|----------------------|
| 0     | MBISTs 7, 38         |
| 1     | MBIST 23             |
| 2     | MBISTs 6, 16, 29, 23 |
| 3     | LBIST 12             |
| 4     | LBISTs 13, 8, 11     |
| 5     | LBISTs 19, 6         |

## 56.7.5 Types of BIST sequences

The STCU2 module on this chip supports these types of built-in self-test (BIST) sequences:

| BIST sequence type   | Description                                                                                                                                                                                                                                                                                                                                                                                             |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Online BIST sequence | <p>A BIST sequence that STCU2 executes during runtime at the request of software. Application software configures and initiates execution of the BIST sequence by loading values into STCU2 registers.</p> <p>Use this type of BIST sequence when you want to test the chip hardware but need to minimize the system startup time. Also use it for failure diagnostics and quality control within a</p> |

*Table continues on the next page...*

| BIST sequence type    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                       | manufacturing environment. You can choose the most appropriate point during runtime at which to execute this type of BIST sequence.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Offline BIST sequence | <p>A BIST sequence that STCU2 automatically executes during reset, either when power is first applied to the chip or when the chip resets STCU2. The chip's SSCM module configures and initiates execution of the BIST sequence by loading values into STCU2 registers from Device Configuration Format (DCF) records that are stored in the chip's UTEST flash memory.</p> <p>Use this type of BIST sequence when you want to test the chip hardware before runtime.</p> <p>The location of the DCF record area is chip-specific. The size of the DCF record area is sequence-specific. For the location of the DCF record area, see the chip-specific STCU2 information.</p> |

### 56.7.6 Supported BIST sequences

NXP has validated and therefore supports only specific built-in self-test (BIST) sequences for this chip. For a list of the supported BIST sequences, see the chip-specific STCU2 information.

### 56.7.7 Definition: Default offline BIST sequence

A *default offline built-in self-test (BIST) sequence* is an offline BIST sequence whose configuration NXP has programmed into the chip's Device Configuration Format (DCF) records that are stored in the chip's UTEST flash memory. To determine whether this chip has a default offline BIST sequence, see the chip-specific STCU2 information.

If this chip has a default offline BIST sequence and you choose to use it, or if it's the only one supported by NXP, there is no need to create additional DCF records for the offline BIST sequence.

## 56.8 Functional description

## 56.8.1 FSM description

The module has three state machines that work together: Master State Machine, Loader/Shifter State Machine, and the Watchdog State Machine. Basically the Master State Machine is the core unit of the STCU2 module. It coordinates all the Self Test operations and the other State Machines. The Loader/Shifter State Machine is used to program the MBIST and the LBIST parameters and to retrieve the related results depending on the parameters stored into the STCU2 registers and under the control of the Master State Machine. The Watchdog State Machine evaluates all the schedule time for MBIST and LBIST and the time-out in case of wrong STCU2 programming.

## 56.8.2 Reset management

Assertion of the STCU2 reset input signal initializes the STCU2 module status, forcing the Off-Line Self-Test condition and cleaning up all the registers and state machine of the module.

The reset signals generated by the STCU2 module at the end of the Self-Test execution phase depends on the Self-Test conditions and on the content of the LBRMSW register.

### 56.8.2.1 Off-Line reset generation

At the end of the Off-Line Self-Test execution the STCU2 raises the global functional reset.

### 56.8.2.2 On-Line reset generation

See the chip-specific STCU2 information for details about On-Line reset generation.

## 56.8.3 Built-in self-test scheduling

The STCU2 module has been designed to be very flexible allowing to program the parallel/serial execution of the MBIST or LBIST depending on the power/timing/coverage constraints. The limitation in the programming flexibility is described in [Design implementation information](#).

The mechanism used to provide this flexibility is a linked list where the starting pointer is the bitfield STCU2\_CFG.PTR. The first LBIST is mapped on 00h, the second on 01h, and so on. The first MBIST is mapped to the address 10h, the second to 11h and so on.

The additional pointers are in the bitfield STCU2\_LB\_CTRL[X].PTR for LBIST[X] (where X is the selected LBIST) and STCU2\_MB\_CTRL[Y].PTR for MBIST (where Y is the selected MBIST) and have to be filled depending on the selected sequence of run. The additional bitfield STCU2\_LB\_CTRL.CSM and STCU2\_MB\_CTRL.CSM provide the flexibility to run concurrently or sequentially the chosen set of the LBIST or MBIST or to close the linked list setting the NIL pointer. For more details, see [STCU2 LBIST Control \(STCU2\\_LB\\_CTRL<sub>n</sub>\)](#) and [STCU2 MBIST Control Register \(STCU2\\_MB\\_CTRL<sub>n</sub>\)](#).

### NOTE

Program operation should ensure that no flash write/erase operations are ongoing when online LBIST is triggered. Otherwise, a flash segment may be corrupted.

## 56.8.4 PLL interface

The STCU2 module is able to directly control the PLL during the Off-Line Self-Test operations depending on the status of the STCU2\_RUN[MBPLEN] and STCU2\_RUN[LBPLEN].

When STCU2\_RUN[RUN] is set to start the Off-Line Self-test operations, the STCU2 takes control of the PLL and changes the PLL clock configurations parameters according to the values programmed into the STCU2\_PLL\_CFG register.

When the STCU2\_RUN[MBPLEN] is active and the MBIST is currently selected, the STCU2 waits for the PLL lock signal. As soon as the lock happens, the STCU2 forces the clock to switch to the PLL output clock. This allows the MBIST to proceed at the PLL output frequency. At the end of the MBIST run the clock source is reverted to the original one.

Same behavior will be applied when STCU2\_RUN[LBPLEN] is active and LBIST is currently selected.

## 56.8.5 Interrupt interface

Interrupt source:

- Triggered by MBIST (enabled by programming STCU2\_RUNSW[MBIE] = 1)

As described in the [RUNSW](#) description, the interrupts are generated at the end of any concurrent run. This means that in case of multiple sequential run in the same session, the number of interrupts generated is equivalent to the number of the sequential session providing in such a way a fine granularity with respect to the completed run.

The MBIFLG bit in the [STCU2 Interrupt Flag Register \(STCU2\\_INT\\_FLG\)](#) register flag interrupt conditions and allow the clearing of interrupt requests. Interrupt bits are cleared by writing the related flag bit to 1 to enable the next interrupt request.

## **56.8.6 FCCU interface**

The FCCU interface is the hardware flag mechanism towards the system, to indicate the occurrence of an Unrecoverable fault and/or a Recoverable fault failure during the Self Test sequence.

To diagnose physical defects on the two fault signals, a fault injection mechanism is also provided. In this case, the FCCU interface allows the user application to check the integrity of the UF and RF connection lines between the STCU2 and the FCCU. Refer to the description of FCCU fault injection mechanism to understand how the UF/RF set/clear mechanism works.

## **56.8.7 Watchdogs**

The STCU2 implements three different watchdogs to ensure that operations are finished in time.

### **56.8.7.1 Initialization watchdog timer**

The STCU2 has two initialization operating modes: program the self-test sequence and run it (Safety mode) or bypass completely the self-test sequence setting STCU2\_RUN[BYP]. In case there are faults during the initialization phase preventing the selection of one of these two operating modes, a hard-coded watchdog time-out flags the wrong behavior.

### **56.8.7.2 Built-in self-test watchdog timer**

The LBIST and MBIST execution time has to be configured as described in [STCU2 Watchdog Granularity \(STCU2\\_WDG\)](#) to account for the overall execution time of the Self-Test sequence. In case the selected LBISTs or MBISTs are not yet completed during



the Off-line Self-Test assigned time, the current LBISTs or MBISTs execution is interrupted and a failure is flagged into STCU2\_ERR\_STAT.WDTO and STCU2\_MBEL/M/H or STCU2\_LBE registers while in case of On-Line Self-Test into STCU2\_ERR\_STAT.WDTOSW and STCU2\_MBELSW/MSW/HSW or STCU2\_LBESW.

In case of multiple sequential run in the same On-LineOff-Line session and the time-out happens in the middle of a sequential run, the next sequential run will be skipped and the execution ends with the current updated status of the registers reported above.

In all the cases, the STCU2 reset generation is managed in one of the modes described in [Reset management](#).

### 56.8.7.3 Register write-access watchdog timer

As explained in the STCU2\_SKC register description:

- A key mechanism protects STCU2 registers during the self-test configuration phase (both online/offline) by preventing any unwanted access.
- A hardware watchdog timer locks register-write access after a number of STCU2 clock cycles. To refresh the hardware watchdog timer before it times out, write only security key 2 to STCU2\_SKC.

AUTOLOCK\_VALUE is the number of STCU2 clock cycles after which the hardware watchdog timer locks register-write access. See the chip-specific STCU2 information for the value of AUTOLOCK\_VALUE.

#### NOTE

For write completion, each DCF record requires a number of STCU2 clock cycles equal to DCF\_COMPLETION. The maximum number of DCF records that can be executed before the watchdog timer must be refreshed is AUTOLOCK\_VALUE divided by DCF\_COMPLETION. See the chip-specific STCU2 information for the value of DCF\_COMPLETION.

The hardware watchdog timer is particularly useful in case STCU2\_CFG[WRP] is 0 during the software self-test configuration. In this case, the software application might enable write access to the STCU2 registers.

## 56.8.8 SSCM interface

The SSCM interface is used to program the STCU2's configuration parameters without the CPU intervention after a reset trigger event initializes the STCU2. This bus interface has the priority over the IPS.

## 56.9 Register description

During Offline SELFTEST, once the STCU2 registers are configured they cannot be overridden.

During Online SELFTEST, once the STCU2 registers are configured they cannot be overridden via IPS till the selftest is complete.

The STCU2 registers are listed in this section.

**STCU2 memory map**

| Address offset (hex) | Register name                                         | Width (in bits) | Access | Reset value                 | Section/ page                |
|----------------------|-------------------------------------------------------|-----------------|--------|-----------------------------|------------------------------|
| 0                    | STCU2 Run Register (STCU2_RUN)                        | 32              | R/W    | 0000_0000h                  | <a href="#">56.9.1/2589</a>  |
| 4                    | STCU2 Run Software Register (STCU2_RUNSW)             | 32              | R/W    | 0000_0000h                  | <a href="#">56.9.2/2590</a>  |
| 8                    | STCU2 SK Code Register (STCU2_SKC)                    | 32              | W      | 0000_0000h                  | <a href="#">56.9.3/2591</a>  |
| C                    | STCU2 Configuration Register (STCU2_CFG)              | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.4/2593</a>  |
| 10                   | STCU2 PLL Configuration Register (STCU2_PLL_CFG)      | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.5/2596</a>  |
| 14                   | STCU2 Watchdog Granularity (STCU2_WDG)                | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.6/2597</a>  |
| 18                   | STCU2 Interrupt Flag Register (STCU2_INT_FLG)         | 32              | w1c    | 0000_0000h                  | <a href="#">56.9.7/2599</a>  |
| 24                   | STCU2 Error Register (STCU2_ERR_STAT)                 | 32              | R/W    | 0000_0000h                  | <a href="#">56.9.8/2600</a>  |
| 28                   | STCU2 Error FM Register (STCU2_ERR_FM)                | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.9/2603</a>  |
| 2C                   | STCU2 Off-Line LBIST Status Register (STCU2_LBS)      | 32              | R      | 0000_0000h                  | <a href="#">56.9.10/2604</a> |
| 30                   | STCU2 Off-Line LBIST End Flag Register (STCU2_LBE)    | 32              | R      | 0000_0000h                  | <a href="#">56.9.11/2605</a> |
| 34                   | STCU2 On-Line LBIST Status Register (STCU2_LBSSW)     | 32              | R      | 0000_0000h                  | <a href="#">56.9.12/2606</a> |
| 38                   | STCU2 On-Line LBIST End Flag Register (STCU2_LBESW)   | 32              | R      | 0000_0000h                  | <a href="#">56.9.13/2608</a> |
| 3C                   | STCU2 On-Line LBIST Reset Management (STCU2_LBRMSW)   | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.14/2609</a> |
| 40                   | STCU2 LBIST Unrecoverable FM Register (STCU2_LBUFM)   | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.15/2611</a> |
| 44                   | STCU2 Off-Line MBIST Status Low Register (STCU2_MBSL) | 32              | R      | 0000_0000h                  | <a href="#">56.9.16/2612</a> |

*Table continues on the next page...*

## STCU2 memory map (continued)

| Address offset (hex) | Register name                                                        | Width (in bits) | Access | Reset value                 | Section/ page                 |
|----------------------|----------------------------------------------------------------------|-----------------|--------|-----------------------------|-------------------------------|
| 48                   | STCU2 Off-Line MBIST Status Medium Register (STCU2_MBSM)             | 32              | R      | 0000_0000h                  | <a href="#">56.9.17/ 2617</a> |
| 4C                   | STCU2 Off-Line MBIST Status High Register (STCU2_MBSH)               | 32              | R      | 0000_0000h                  | <a href="#">56.9.18/ 2622</a> |
| 50                   | STCU2 Off-Line MBIST End Flag Low Register (STCU2_MBEL)              | 32              | R      | 0000_0000h                  | <a href="#">56.9.19/ 2623</a> |
| 54                   | STCU2 Off-Line MBIST End Flag Medium Register (STCU2_MBEM)           | 32              | R      | 0000_0000h                  | <a href="#">56.9.20/ 2627</a> |
| 58                   | STCU2 Off-Line MBIST End Flag High Register (STCU2_MBEH)             | 32              | R      | 0000_0000h                  | <a href="#">56.9.21/ 2631</a> |
| 5C                   | STCU2 On-Line MBIST Status Low Register (STCU2_MBSLSW)               | 32              | R      | 0000_0000h                  | <a href="#">56.9.22/ 2632</a> |
| 60                   | STCU2 On-Line MBIST Status Medium Register (STCU2_MBSMSW)            | 32              | R      | 0000_0000h                  | <a href="#">56.9.23/ 2635</a> |
| 64                   | STCU2 On-Line MBIST Status High Register (STCU2_MBSHSW)              | 32              | R      | 0000_0000h                  | <a href="#">56.9.24/ 2639</a> |
| 68                   | STCU2 On-Line MBIST End Flag Low Register (STCU2_MBELSW)             | 32              | R      | 0000_0000h                  | <a href="#">56.9.25/ 2640</a> |
| 6C                   | STCU2 On-Line MBIST End Flag Medium Register (STCU2_MBEMSW)          | 32              | R      | 0000_0000h                  | <a href="#">56.9.26/ 2644</a> |
| 70                   | STCU2 On-Line MBIST End Flag High Register (STCU2_MBEHSW)            | 32              | R      | 0000_0000h                  | <a href="#">56.9.27/ 2648</a> |
| 74                   | STCU2 MBIST Unrecoverable FM Low Register (STCU2_MBUFML)             | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.28/ 2649</a> |
| 78                   | STCU2 MBIST Unrecoverable FM Medium Register (STCU2_MBUFMM)          | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.29/ 2652</a> |
| 7C                   | STCU2 MBIST Unrecoverable FM High Register (STCU2_MBUFMH)            | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.30/ 2656</a> |
| 100                  | STCU2 LBIST Control (STCU2_LB_CTRL0)                                 | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.31/ 2657</a> |
| 104                  | STCU2 LBIST PC Stop Register (STCU2_LB_PCS0)                         | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.32/ 2660</a> |
| 110                  | STCU2 Off-Line LBIST MISR Expected Low Register (STCU2_LB_MISREL0)   | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.33/ 2660</a> |
| 114                  | STCU2 Off-Line LBIST MISR Expected High Register (STCU2_LB_MISREH0)  | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.34/ 2661</a> |
| 118                  | STCU2 Off-Line LBIST MISR Read Low Register (STCU2_LB_MISRRL0)       | 32              | R      | <a href="#">See section</a> | <a href="#">56.9.35/ 2662</a> |
| 11C                  | STCU2 Off-Line LBIST MISR Read High Register (STCU2_LB_MISRHH0)      | 32              | R      | <a href="#">See section</a> | <a href="#">56.9.36/ 2662</a> |
| 120                  | STCU2 On-Line LBIST MISR Expected Low Register (STCU2_LB_MISRELSW0)  | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.37/ 2663</a> |
| 124                  | STCU2 On-Line LBIST MISR Expected High Register (STCU2_LB_MISREHSW0) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.38/ 2664</a> |

Table continues on the next page...

## STCU2 memory map (continued)

| Address offset (hex) | Register name                                                        | Width (in bits) | Access | Reset value | Section/ page    |
|----------------------|----------------------------------------------------------------------|-----------------|--------|-------------|------------------|
| 128                  | STCU2 On-Line LBIST MISR Read Low Register (STCU2_LB_MISRRLSW0)      | 32              | R      | See section | 56.9.39/<br>2664 |
| 12C                  | STCU2 On-Line LBIST MISR Read High Register (STCU2_LB_MISRHRHSW0)    | 32              | R      | See section | 56.9.40/<br>2665 |
| 140                  | STCU2 LBIST Control (STCU2_LB_CTRL1)                                 | 32              | R/W    | See section | 56.9.31/<br>2657 |
| 144                  | STCU2 LBIST PC Stop Register (STCU2_LB_PCS1)                         | 32              | R/W    | See section | 56.9.32/<br>2660 |
| 150                  | STCU2 Off-Line LBIST MISR Expected Low Register (STCU2_LB_MISREL1)   | 32              | R/W    | See section | 56.9.33/<br>2660 |
| 154                  | STCU2 Off-Line LBIST MISR Expected High Register (STCU2_LB_MISREH1)  | 32              | R/W    | See section | 56.9.34/<br>2661 |
| 158                  | STCU2 Off-Line LBIST MISR Read Low Register (STCU2_LB_MISRRL1)       | 32              | R      | See section | 56.9.35/<br>2662 |
| 15C                  | STCU2 Off-Line LBIST MISR Read High Register (STCU2_LB_MISRHRH1)     | 32              | R      | See section | 56.9.36/<br>2662 |
| 160                  | STCU2 On-Line LBIST MISR Expected Low Register (STCU2_LB_MISRELSW1)  | 32              | R/W    | See section | 56.9.37/<br>2663 |
| 164                  | STCU2 On-Line LBIST MISR Expected High Register (STCU2_LB_MISREHSW1) | 32              | R/W    | See section | 56.9.38/<br>2664 |
| 168                  | STCU2 On-Line LBIST MISR Read Low Register (STCU2_LB_MISRRLSW1)      | 32              | R      | See section | 56.9.39/<br>2664 |
| 16C                  | STCU2 On-Line LBIST MISR Read High Register (STCU2_LB_MISRHRHSW1)    | 32              | R      | See section | 56.9.40/<br>2665 |
| 180                  | STCU2 LBIST Control (STCU2_LB_CTRL2)                                 | 32              | R/W    | See section | 56.9.31/<br>2657 |
| 184                  | STCU2 LBIST PC Stop Register (STCU2_LB_PCS2)                         | 32              | R/W    | See section | 56.9.32/<br>2660 |
| 190                  | STCU2 Off-Line LBIST MISR Expected Low Register (STCU2_LB_MISREL2)   | 32              | R/W    | See section | 56.9.33/<br>2660 |
| 194                  | STCU2 Off-Line LBIST MISR Expected High Register (STCU2_LB_MISREH2)  | 32              | R/W    | See section | 56.9.34/<br>2661 |
| 198                  | STCU2 Off-Line LBIST MISR Read Low Register (STCU2_LB_MISRRL2)       | 32              | R      | See section | 56.9.35/<br>2662 |
| 19C                  | STCU2 Off-Line LBIST MISR Read High Register (STCU2_LB_MISRHRH2)     | 32              | R      | See section | 56.9.36/<br>2662 |
| 1A0                  | STCU2 On-Line LBIST MISR Expected Low Register (STCU2_LB_MISRELSW2)  | 32              | R/W    | See section | 56.9.37/<br>2663 |
| 1A4                  | STCU2 On-Line LBIST MISR Expected High Register (STCU2_LB_MISREHSW2) | 32              | R/W    | See section | 56.9.38/<br>2664 |
| 1A8                  | STCU2 On-Line LBIST MISR Read Low Register (STCU2_LB_MISRRLSW2)      | 32              | R      | See section | 56.9.39/<br>2664 |
| 1AC                  | STCU2 On-Line LBIST MISR Read High Register (STCU2_LB_MISRHRHSW2)    | 32              | R      | See section | 56.9.40/<br>2665 |

Table continues on the next page...

## STCU2 memory map (continued)

| Address offset (hex) | Register name                                                        | Width (in bits) | Access | Reset value | Section/ page    |
|----------------------|----------------------------------------------------------------------|-----------------|--------|-------------|------------------|
| 1C0                  | STCU2 LBIST Control (STCU2_LB_CTRL3)                                 | 32              | R/W    | See section | 56.9.31/<br>2657 |
| 1C4                  | STCU2 LBIST PC Stop Register (STCU2_LB_PCS3)                         | 32              | R/W    | See section | 56.9.32/<br>2660 |
| 1D0                  | STCU2 Off-Line LBIST MISR Expected Low Register (STCU2_LB_MISREL3)   | 32              | R/W    | See section | 56.9.33/<br>2660 |
| 1D4                  | STCU2 Off-Line LBIST MISR Expected High Register (STCU2_LB_MISREH3)  | 32              | R/W    | See section | 56.9.34/<br>2661 |
| 1D8                  | STCU2 Off-Line LBIST MISR Read Low Register (STCU2_LB_MISRRL3)       | 32              | R      | See section | 56.9.35/<br>2662 |
| 1DC                  | STCU2 Off-Line LBIST MISR Read High Register (STCU2_LB_MISRRH3)      | 32              | R      | See section | 56.9.36/<br>2662 |
| 1E0                  | STCU2 On-Line LBIST MISR Expected Low Register (STCU2_LB_MISRELSW3)  | 32              | R/W    | See section | 56.9.37/<br>2663 |
| 1E4                  | STCU2 On-Line LBIST MISR Expected High Register (STCU2_LB_MISREHSW3) | 32              | R/W    | See section | 56.9.38/<br>2664 |
| 1E8                  | STCU2 On-Line LBIST MISR Read Low Register (STCU2_LB_MISRRLSW3)      | 32              | R      | See section | 56.9.39/<br>2664 |
| 1EC                  | STCU2 On-Line LBIST MISR Read High Register (STCU2_LB_MISRRHSW3)     | 32              | R      | See section | 56.9.40/<br>2665 |
| 200                  | STCU2 LBIST Control (STCU2_LB_CTRL4)                                 | 32              | R/W    | See section | 56.9.31/<br>2657 |
| 204                  | STCU2 LBIST PC Stop Register (STCU2_LB_PCS4)                         | 32              | R/W    | See section | 56.9.32/<br>2660 |
| 210                  | STCU2 Off-Line LBIST MISR Expected Low Register (STCU2_LB_MISREL4)   | 32              | R/W    | See section | 56.9.33/<br>2660 |
| 214                  | STCU2 Off-Line LBIST MISR Expected High Register (STCU2_LB_MISREH4)  | 32              | R/W    | See section | 56.9.34/<br>2661 |
| 218                  | STCU2 Off-Line LBIST MISR Read Low Register (STCU2_LB_MISRRL4)       | 32              | R      | See section | 56.9.35/<br>2662 |
| 21C                  | STCU2 Off-Line LBIST MISR Read High Register (STCU2_LB_MISRRH4)      | 32              | R      | See section | 56.9.36/<br>2662 |
| 220                  | STCU2 On-Line LBIST MISR Expected Low Register (STCU2_LB_MISRELSW4)  | 32              | R/W    | See section | 56.9.37/<br>2663 |
| 224                  | STCU2 On-Line LBIST MISR Expected High Register (STCU2_LB_MISREHSW4) | 32              | R/W    | See section | 56.9.38/<br>2664 |
| 228                  | STCU2 On-Line LBIST MISR Read Low Register (STCU2_LB_MISRRLSW4)      | 32              | R      | See section | 56.9.39/<br>2664 |
| 22C                  | STCU2 On-Line LBIST MISR Read High Register (STCU2_LB_MISRRHSW4)     | 32              | R      | See section | 56.9.40/<br>2665 |
| 240                  | STCU2 LBIST Control (STCU2_LB_CTRL5)                                 | 32              | R/W    | See section | 56.9.31/<br>2657 |
| 244                  | STCU2 LBIST PC Stop Register (STCU2_LB_PCS5)                         | 32              | R/W    | See section | 56.9.32/<br>2660 |

Table continues on the next page...

## STCU2 memory map (continued)

| Address offset (hex) | Register name                                                        | Width (in bits) | Access | Reset value | Section/ page    |
|----------------------|----------------------------------------------------------------------|-----------------|--------|-------------|------------------|
| 250                  | STCU2 Off-Line LBIST MISR Expected Low Register (STCU2_LB_MISREL5)   | 32              | R/W    | See section | 56.9.33/<br>2660 |
| 254                  | STCU2 Off-Line LBIST MISR Expected High Register (STCU2_LB_MISREH5)  | 32              | R/W    | See section | 56.9.34/<br>2661 |
| 258                  | STCU2 Off-Line LBIST MISR Read Low Register (STCU2_LB_MISRRL5)       | 32              | R      | See section | 56.9.35/<br>2662 |
| 25C                  | STCU2 Off-Line LBIST MISR Read High Register (STCU2_LB_MISRRLH5)     | 32              | R      | See section | 56.9.36/<br>2662 |
| 260                  | STCU2 On-Line LBIST MISR Expected Low Register (STCU2_LB_MISRELSW5)  | 32              | R/W    | See section | 56.9.37/<br>2663 |
| 264                  | STCU2 On-Line LBIST MISR Expected High Register (STCU2_LB_MISREHSW5) | 32              | R/W    | See section | 56.9.38/<br>2664 |
| 268                  | STCU2 On-Line LBIST MISR Read Low Register (STCU2_LB_MISRRLSW5)      | 32              | R      | See section | 56.9.39/<br>2664 |
| 26C                  | STCU2 On-Line LBIST MISR Read High Register (STCU2_LB_MISRRLHSW5)    | 32              | R      | See section | 56.9.40/<br>2665 |
| 600                  | STCU2 MBIST Control Register (STCU2_MB_CTRL0)                        | 32              | R/W    | See section | 56.9.41/<br>2666 |
| 604                  | STCU2 MBIST Control Register (STCU2_MB_CTRL1)                        | 32              | R/W    | See section | 56.9.41/<br>2666 |
| 608                  | STCU2 MBIST Control Register (STCU2_MB_CTRL2)                        | 32              | R/W    | See section | 56.9.41/<br>2666 |
| 60C                  | STCU2 MBIST Control Register (STCU2_MB_CTRL3)                        | 32              | R/W    | See section | 56.9.41/<br>2666 |
| 610                  | STCU2 MBIST Control Register (STCU2_MB_CTRL4)                        | 32              | R/W    | See section | 56.9.41/<br>2666 |
| 614                  | STCU2 MBIST Control Register (STCU2_MB_CTRL5)                        | 32              | R/W    | See section | 56.9.41/<br>2666 |
| 618                  | STCU2 MBIST Control Register (STCU2_MB_CTRL6)                        | 32              | R/W    | See section | 56.9.41/<br>2666 |
| 61C                  | STCU2 MBIST Control Register (STCU2_MB_CTRL7)                        | 32              | R/W    | See section | 56.9.41/<br>2666 |
| 620                  | STCU2 MBIST Control Register (STCU2_MB_CTRL8)                        | 32              | R/W    | See section | 56.9.41/<br>2666 |
| 624                  | STCU2 MBIST Control Register (STCU2_MB_CTRL9)                        | 32              | R/W    | See section | 56.9.41/<br>2666 |
| 628                  | STCU2 MBIST Control Register (STCU2_MB_CTRL10)                       | 32              | R/W    | See section | 56.9.41/<br>2666 |
| 62C                  | STCU2 MBIST Control Register (STCU2_MB_CTRL11)                       | 32              | R/W    | See section | 56.9.41/<br>2666 |
| 630                  | STCU2 MBIST Control Register (STCU2_MB_CTRL12)                       | 32              | R/W    | See section | 56.9.41/<br>2666 |
| 634                  | STCU2 MBIST Control Register (STCU2_MB_CTRL13)                       | 32              | R/W    | See section | 56.9.41/<br>2666 |

Table continues on the next page...



**STCU2 memory map (continued)**

| Address offset (hex) | Register name                                  | Width (in bits) | Access | Reset value                 | Section/page                 |
|----------------------|------------------------------------------------|-----------------|--------|-----------------------------|------------------------------|
| 638                  | STCU2 MBIST Control Register (STCU2_MB_CTRL14) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 63C                  | STCU2 MBIST Control Register (STCU2_MB_CTRL15) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 640                  | STCU2 MBIST Control Register (STCU2_MB_CTRL16) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 644                  | STCU2 MBIST Control Register (STCU2_MB_CTRL17) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 648                  | STCU2 MBIST Control Register (STCU2_MB_CTRL18) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 64C                  | STCU2 MBIST Control Register (STCU2_MB_CTRL19) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 650                  | STCU2 MBIST Control Register (STCU2_MB_CTRL20) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 654                  | STCU2 MBIST Control Register (STCU2_MB_CTRL21) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 658                  | STCU2 MBIST Control Register (STCU2_MB_CTRL22) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 65C                  | STCU2 MBIST Control Register (STCU2_MB_CTRL23) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 660                  | STCU2 MBIST Control Register (STCU2_MB_CTRL24) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 664                  | STCU2 MBIST Control Register (STCU2_MB_CTRL25) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 668                  | STCU2 MBIST Control Register (STCU2_MB_CTRL26) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 66C                  | STCU2 MBIST Control Register (STCU2_MB_CTRL27) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 670                  | STCU2 MBIST Control Register (STCU2_MB_CTRL28) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 674                  | STCU2 MBIST Control Register (STCU2_MB_CTRL29) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 678                  | STCU2 MBIST Control Register (STCU2_MB_CTRL30) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 67C                  | STCU2 MBIST Control Register (STCU2_MB_CTRL31) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 680                  | STCU2 MBIST Control Register (STCU2_MB_CTRL32) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 684                  | STCU2 MBIST Control Register (STCU2_MB_CTRL33) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 688                  | STCU2 MBIST Control Register (STCU2_MB_CTRL34) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 68C                  | STCU2 MBIST Control Register (STCU2_MB_CTRL35) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |

*Table continues on the next page...*

## STCU2 memory map (continued)

| Address offset (hex) | Register name                                  | Width (in bits) | Access | Reset value                 | Section/page                 |
|----------------------|------------------------------------------------|-----------------|--------|-----------------------------|------------------------------|
| 690                  | STCU2 MBIST Control Register (STCU2_MB_CTRL36) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 694                  | STCU2 MBIST Control Register (STCU2_MB_CTRL37) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 698                  | STCU2 MBIST Control Register (STCU2_MB_CTRL38) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 69C                  | STCU2 MBIST Control Register (STCU2_MB_CTRL39) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 6A0                  | STCU2 MBIST Control Register (STCU2_MB_CTRL40) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 6A4                  | STCU2 MBIST Control Register (STCU2_MB_CTRL41) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 6A8                  | STCU2 MBIST Control Register (STCU2_MB_CTRL42) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 6AC                  | STCU2 MBIST Control Register (STCU2_MB_CTRL43) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 6B0                  | STCU2 MBIST Control Register (STCU2_MB_CTRL44) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 6B4                  | STCU2 MBIST Control Register (STCU2_MB_CTRL45) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 6B8                  | STCU2 MBIST Control Register (STCU2_MB_CTRL46) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 6BC                  | STCU2 MBIST Control Register (STCU2_MB_CTRL47) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 6C0                  | STCU2 MBIST Control Register (STCU2_MB_CTRL48) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 6C4                  | STCU2 MBIST Control Register (STCU2_MB_CTRL49) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 6C8                  | STCU2 MBIST Control Register (STCU2_MB_CTRL50) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 6CC                  | STCU2 MBIST Control Register (STCU2_MB_CTRL51) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 6D0                  | STCU2 MBIST Control Register (STCU2_MB_CTRL52) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 6D4                  | STCU2 MBIST Control Register (STCU2_MB_CTRL53) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 6D8                  | STCU2 MBIST Control Register (STCU2_MB_CTRL54) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 6DC                  | STCU2 MBIST Control Register (STCU2_MB_CTRL55) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 6E0                  | STCU2 MBIST Control Register (STCU2_MB_CTRL56) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |
| 6E4                  | STCU2 MBIST Control Register (STCU2_MB_CTRL57) | 32              | R/W    | <a href="#">See section</a> | <a href="#">56.9.41/2666</a> |

Table continues on the next page...



## STCU2 memory map (continued)

| Address offset (hex) | Register name                                  | Width (in bits) | Access | Reset value | Section/page     |
|----------------------|------------------------------------------------|-----------------|--------|-------------|------------------|
| 6E8                  | STCU2 MBIST Control Register (STCU2_MB_CTRL58) | 32              | R/W    | See section | 56.9.41/<br>2666 |
| 6EC                  | STCU2 MBIST Control Register (STCU2_MB_CTRL59) | 32              | R/W    | See section | 56.9.41/<br>2666 |
| 6F0                  | STCU2 MBIST Control Register (STCU2_MB_CTRL60) | 32              | R/W    | See section | 56.9.41/<br>2666 |
| 6F4                  | STCU2 MBIST Control Register (STCU2_MB_CTRL61) | 32              | R/W    | See section | 56.9.41/<br>2666 |
| 6F8                  | STCU2 MBIST Control Register (STCU2_MB_CTRL62) | 32              | R/W    | See section | 56.9.41/<br>2666 |
| 6FC                  | STCU2 MBIST Control Register (STCU2_MB_CTRL63) | 32              | R/W    | See section | 56.9.41/<br>2666 |
| 700                  | STCU2 MBIST Control Register (STCU2_MB_CTRL64) | 32              | R/W    | See section | 56.9.41/<br>2666 |

## 56.9.1 STCU2 Run Register (STCU2\_RUN)

The STCU2\_RUN register defines the RUN bit to start the off-line self testing procedure.

The R/W fields in this register are readable at any time. However, you can write to these fields only when Off-line Self-Test phase is still active.

Address: 0h base + 0h offset = 0h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |    |    |    |    |     |        |        |    |    |    |    |    |    |    |     |
|-------|----|----|----|----|----|-----|--------|--------|----|----|----|----|----|----|----|-----|
| Bit   | 16 | 17 | 18 | 19 | 20 | 21  | 22     | 23     | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31  |
| R     | 0  |    |    |    |    |     |        |        | 0  |    |    |    |    |    |    |     |
| W     |    |    |    |    |    | BYP | MBPLEN | LBPLEN |    |    |    |    |    |    |    | RUN |
| Reset | 0  | 0  | 0  | 0  | 0  | 0   | 0      | 0      | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   |

## STCU2\_RUN field descriptions

| Field            | Description                                                                             |
|------------------|-----------------------------------------------------------------------------------------|
| 0–20<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |

Table continues on the next page...

**STCU2\_RUN field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                    |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 21<br>BYP         | Bypass mode<br><br>0 Off-Line Self-Test is active. In case the RUN bit into the STCU2_RUN register is not set before the Hard-Coded WDG Time-out, the WDTO error is generated into the STCU2_ERR_STAT register.<br>1 Off-Line Self-Test is completely bypassed and the access to STCU2 is locked until the STCU2_SKC register is written according to the request access mode. |
| 22<br>MBPLEN      | Off-Line MBIST with PLL Enabled<br><br>0 Off-Line MBIST is executed without using the on-chip PLL.<br>1 Off-Line MBIST is executed enabling the on-chip PLL control interface selecting the parameters defined into STCU2_PLL_CFG register.                                                                                                                                    |
| 23<br>LBPLEN      | Off-Line LBIST with PLL Enabled<br><br>0 Off-Line LBIST is executed without using the on-chip PLL.<br>1 Off-Line LBIST is executed enabling the on-chip PLL control interface selecting the parameters defined into STCU2_PLL_CFG register.                                                                                                                                    |
| 24–30<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                        |
| 31<br>RUN         | RUN: RUN<br><br>The RUN bit is automatically cleared by the STCU2 when the off-line self testing procedure has been completed.<br><br>0 Idle<br>1 Off-Line Self testing procedure is running                                                                                                                                                                                   |

**56.9.2 STCU2 Run Software Register (STCU2\_RUNSW)**

The STCU2\_RUNSW register defines the RUN bit to start the on-line self testing procedure.

The R/W fields in this register are readable at any time. However, you can write to these fields only when both of the following conditions are true:

- STCU2\_CFG[WRP] = 0
- On-line Self-Test phase is active. The On-Line condition starts when the Off-Line is over. In this condition, system is alive and the SW can program the STCU2

Address: 0h base + 4h offset = 4h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |    |    |    |      |    |          |          |    |    |    |    |    |    |    |    |
|-------|----|----|----|----|------|----|----------|----------|----|----|----|----|----|----|----|----|
| Bit   | 16 | 17 | 18 | 19 | 20   | 21 | 22       | 23       | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0  |    |    |    | MBIE | 0  | MBSWPLEN | LBSWPLEN | 0  |    |    |    | 0  | 0  | 0  |    |
| W     |    |    |    |    |      |    |          |          |    |    |    |    |    |    |    |    |
| Reset | 0  | 0  | 0  | 0  | 0    | 0  | 0        | 0        | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

### STCU2\_RUNSW field descriptions

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–19<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 20<br>MBIE        | MBIST Interrupt Enable<br><br>0 Interrupt is not generated at the end of the software MBIST execution phase of the selected Memories.<br>1 At the end of the software MBIST execution phase of the concurrent selected memories the STCU2 forces an interrupt request to notify that MBISTs of concurrent selected memories are over. In case there is a sequence of concurrent runs in the same Self-Test run, the interrupt request is generated at the end of each concurrent runs provided that the software will clear the STCU2_INT_FLG[MBIFG] bit in between. |
| 21<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 22<br>MBSWPLEN    | On-Line MBIST with PLL Enabled<br><br>0 On-Line MBIST is executed without using the on-chip PLL.<br>1 On-Line MBIST is executed using the PLL configuration provided by software. STCU2 does not take the PLL control but monitors the PLL lock signal to check if PLL is working correctly.                                                                                                                                                                                                                                                                         |
| 23<br>LBSWPLEN    | On-Line LBIST with PLL Enabled<br><br>0 On-Line LBIST is executed without using the on-chip PLL.<br>1 On-Line LBIST is executed using the PLL configuration provided by software. STCU2 does not take the PLL control but monitors the PLL lock signal to check if PLL is working correctly.                                                                                                                                                                                                                                                                         |
| 24–29<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 30<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 31<br>RUNSW       | RUNSW<br>The RUNSW bit is automatically cleared by the STCU2 when the on-line self testing procedure has been completed.<br><br>0 Idle<br>1 On-line self testing procedure is running                                                                                                                                                                                                                                                                                                                                                                                |

### 56.9.3 STCU2 SK Code Register (STCU2\_SKC)

The STCU2\_SKC register implements the security key code mechanism needed to access in write mode to the other STCU2 registers. In order to unlock the STCU2 access after:

**Register description**

- The STCU2 asynchronous reset
- The end of the STCU2 run

the software (IPS bus) or the SSCM interfaces have to apply the following sequence:

- write the key1 into the STCU2\_SKC register
- write the key2 into the STCU2\_SKC register

Depending on the off/on-line test step, the two keys will be different. Byte write operation is not allowed since the full key has to be recognized as one unit.

It should be noted that after the Self-Test sequence has been completed or the Bypass feature has been enabled (setting the bit BYP into the STCU2\_CFG), the SSCM interface is no longer available.

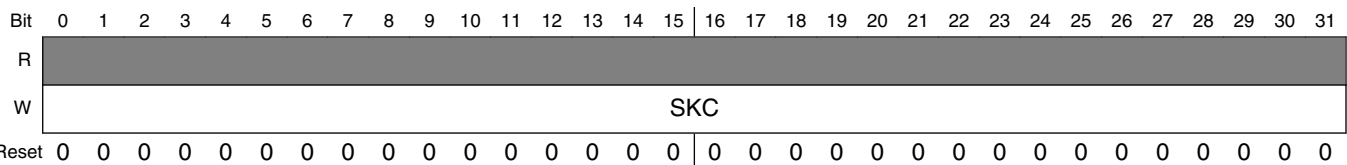
In case of invalid access or sequence (Key1/2 have to be applied consecutively), a transfer error on the IPS or SSCM bus is asserted depending on the selected source. The STCU2 write access is locked and to unlock the access the sequence has to be applied again.

In case the STCU2 register access lasts more cycles than the one defined into the Hard-coded WDG time-out, the STCU2 write access is locked and the WDG and Register ITF clocks are switched off. Also in this case, in order to enable again the write access to the STCU2 and the WDG and Register ITF clocks, it is required to apply again the sequence.

In case it is required to extend the STCU2 register access cycles before the Hard-coded WDG time-out expires, only the Key2 has to be applied. The effect of this write operation is to re-initialize the WDG time-out counter. In such a condition, Key1 has not to be applied otherwise a transfer error on the IPS bus is asserted depending on the selected source. The STCU2 write access is locked and to unlock the access the sequence has to be applied again.

The STCU2\_SKC register is not readable. The value 00000000h is always returned in case of read operation.

Address: 0h base + 8h offset = 8h



**STCU2\_SKC field descriptions**

| Field       | Description                                                                                         |
|-------------|-----------------------------------------------------------------------------------------------------|
| 0-31<br>SKC | STCU2 security key code for off-line Test<br>= D3FEA98Bh: Key1 to unlock the write access the STCU2 |

**STCU2\_SKC field descriptions (continued)**

| Field | Description                                                                                                                                                                                                            |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | = 2C015674h: Key2 to unlock the write access the STCU2<br>STCU2 security key code for on-line Test<br>= 753F924Eh: Key1 to unlock the write access the STCU2<br>= 8AC06DB1h: Key2 to unlock the write access the STCU2 |

**56.9.4 STCU2 Configuration Register (STCU2\_CFG)****NOTE**

The reset value is chip-specific; see the chapter that describes how modules are configured and connected.

The STCU2\_CFG register includes the global configuration of the STCU2 and can be updated both in the Off/On-Line Test steps.

The access to this register is described in the following figure. It further depends on the state of the WRP bit as follows:

- When WRP = 0: The register can be written during the on-line Self-Test case without restrictions
- When WRP = 1:
  - The only bit that can be written without any restriction is WRP.
  - In case there are software operations that write all the register's bits, a transfer error is raised only if the value of the selected byte written differs from the current status of the register. This functionality has been implemented to prevent potential compilation behavior that might invalidate this single bit clean capability.

MBIST can be run in three different test types according to how the STCU2\_CFG register bits 27 and 28 are set.

The following table describes the test algorithm and coverage of the different test types.

**Table 56-10. MBIST algorithm and coverage according to test type**

| MBIST Type | STCU2_CFG[PM OSEN] | STCU2_CFG[MB U] | MBIST Algorithm                                                         | Internal Nodes/ Circuits Additionally Covered by Test                           | Use Case                                                                                 | Test Cycle                        |
|------------|--------------------|-----------------|-------------------------------------------------------------------------|---------------------------------------------------------------------------------|------------------------------------------------------------------------------------------|-----------------------------------|
| Full Test  | 1                  | 0               | Test memory using all built in algorithms. Open PMOS algorithm included | Address Decoder and Bitcell as well as resistive defects in the address decoder | Used by MCU production test. Recommended to use in the field for MBIST online self-test. | About 3 to 4x more than Auto Test |

*Table continues on the next page...*

**Table 56-10. MBIST algorithm and coverage according to test type (continued)**

| MBIST Type   | STCU2_CFG[PM OSEN] | STCU2_CFG[MB U] | MBIST Algorithm                                                           | Internal Nodes/<br>Circuits Additionally Covered by Test            | Use Case                                                                                              | Test Cycle                          |
|--------------|--------------------|-----------------|---------------------------------------------------------------------------|---------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|-------------------------------------|
| Reduced Test | 0                  | 0               | Test memory using all built in algorithms except the open PMOS algorithm. | Address Decoder and Bitcell                                         | Recommended to use in the field for MBIST online self-test if the Full Test takes too long.           | About 10% less than Full Test       |
| Auto Test    | x                  | 1               | A smaller set of algorithms which target latent defect mechanisms.        | Latent defects such as NBTI of the PMOS transistors in the bitcells | Recommended to use for MBIST offline self-test as an optimum balance of test time vs. fault coverage. | About 3 to 4x faster than Full Test |

Address: 0h base + Ch offset = Ch

|       |    |     |    |    |    |    |    |     |          |    |        |     |         |    |    |    |
|-------|----|-----|----|----|----|----|----|-----|----------|----|--------|-----|---------|----|----|----|
| Bit   | 0  | 1   | 2  | 3  | 4  | 5  | 6  | 7   | 8        | 9  | 10     | 11  | 12      | 13 | 14 | 15 |
| R     | 0  | PTR |    |    |    |    |    |     | LB_DELAY |    |        |     |         |    |    |    |
| W     |    |     |    |    |    |    |    |     |          |    |        |     |         |    |    |    |
| Reset | x* | x*  | x* | x* | x* | x* | x* | x*  | x*       | x* | x*     | x*  | x*      | x* | x* | x* |
| Bit   | 16 | 17  | 18 | 19 | 20 | 21 | 22 | 23  | 24       | 25 | 26     | 27  | 28      | 29 | 30 | 31 |
| R     | 0  |     |    |    |    |    |    | WRP | 0        | 0  | PMOSEN | MBU | CLK_CFG |    |    |    |
| W     |    |     |    |    |    |    |    |     |          |    |        |     |         |    |    |    |
| Reset | x* | x*  | x* | x* | x* | x* | x* | x*  | x*       | x* | x*     | x*  | x*      | x* | x* | x* |

\* Notes:

- x = Undefined at reset.

**STCU2\_CFG field descriptions**

| Field         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 1–7<br>PTR    | LBIST or MBIST pointer<br>PTR defines the logical pointer to the first LBIST or MBIST to be scheduled when the self testing procedure is enabled. PTR is the entry pointer to a linked list of BIST descriptors. If PTR = 0 then the LBIST0 is initially scheduled.<br>You can always read this field. In the off-line Self-Test phase, you can always write to this field. In the on-line Self-Test phase, you can write to this field only when WRP = 0.<br>0h to (LBIST -1): pointer to LBIST<br>10h to (10h + MBIST -1): pointer to MBIST<br>7Fh: pointer to NIL. No BIST execution. |

Table continues on the next page...

## STCU2\_CFG field descriptions (continued)

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 8–15<br>LB_DELAY  | <p>Delay LBIST run</p> <p>LB_DELAY defines the delay between the LBIST starts when more than a single LBIST is selected to be executed concurrently with the purpose of smoothing the power consumption transient. The allowed delay time are the following:</p> <p>00h: No delay</p> <p>01h: 1 x 16 STCU2 Core clock cycles</p> <p>02h: 2 x 16 STCU2 Core clock cycles</p> <p>03h: 3 x 16 STCU2 Core clock cycles</p> <p>...</p> <p>FDh: 253 x 16 STCU2 Core clock cycles</p> <p>FEh: 254 x 16 STCU2 Core clock cycles</p> <p>FFh: 255 x 16 STCU2 Core clock cycles</p> <p>Note that in case the Frequency of the STCU2 core clock is 16 MHz, the delay parameters is expressed in <math>\mu</math>s.</p> <p><b>NOTE:</b> This field should be written with 0xFF for self test run.</p> <p>You can always read this field. In the off-line Self-Test phase, you can always write to this field. In the on-line Self-Test phase, you can write to this field only when WRP = 0.</p> |
| 16–22<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 23<br>WRP         | <p>Write Protection</p> <p>0: Specific STCU2 registers can be written through IPS bus interface after Off-Line Self-Test sequence has been completed</p> <p>1: Specific STCU2 registers cannot be written though IPS also after Off-Line Self-Test sequence has been completed preventing in such a way any user application write operation</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 24–25<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 26<br>Reserved    | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 27<br>PMOSEN      | <p>MBIST PMOS Test Enable</p> <p>When MBU is 1 then "Auto test" MBIST test type will be selected regardless of the setting of this field.</p> <p>0 When MBU is 0 then selects the "Reduced test" MBIST test type.</p> <p>1 When MBU is 0 then selects the "Full test" MBIST test type.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 28<br>MBU         | <p>MBIST MBU Test Enabled</p> <p>0 Either "Full test" or "Reduced test" MBIST test type will be selected according to the PMOSEN value.</p> <p>1 Selects "Auto test" MBIST test type regardless of the PMOSEN value.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 29–31<br>CLK_CFG  | <p>Logic, Memory BIST and STCU2 core CLK Clock configuration</p> <p>CLK_CFG defines the ratio between the sys_clk and the TCK used to program both the LBIST and the MBIST and the STCU2 core clock. The punch-out mechanism is used to generate the derived clocks. The allowed configurations are the following:</p> <p>000: sys_clk</p> <p>001: sys_clk/2</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

Table continues on the next page...

## STCU2\_CFG field descriptions (continued)

| Field | Description                                                                                                                                                                                                                                                            |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|       | 010: sys_clk/3<br>...<br>101: sys_clk/6<br>110: sys_clk/7<br>111: sys_clk/8<br><br>You can always read this field. In the off-line Self-Test phase, you can always write to this field. In the on-line Self-Test phase, you can write to this field only when WRP = 0. |

## 56.9.5 STCU2 PLL Configuration Register (STCU2\_PLL\_CFG)

## NOTE

The reset value is chip-specific; see the chapter that describes how modules are configured and connected.

The STCU2\_PLL\_CFG register defines the parameters used to program the PLL only when the Off-line L/MBIST are performed and STCU2\_RUN[MBPLEN] = 1 or STCU2\_RUN[LBPLEN] = 1.

In the On-line condition, these bits are not effective. It is also possible to set the PLL during active On-line mode to prevent any potential functionality issue.

The R/W fields in this register are readable at any time. However, you can write to these fields only when Off-line Self-Test phase is still active.

Address: 0h base + 10h offset = 10h

|       |    |    |        |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |    |    |    |    |    |    |    |        |        |    |    |    |    |    |
|-------|----|----|--------|----|----|----|----|----|----|----|----|----|--------|----|----|----|----|----|----|----|----|----|----|----|----|--------|--------|----|----|----|----|----|
| Bit   | 0  | 1  | 2      | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12     | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25     | 26     | 27 | 28 | 29 | 30 | 31 |
| R     | 0  |    |        |    |    |    |    |    | 0  |    |    |    | PLLIDF |    |    | 0  |    |    |    |    |    |    |    |    |    | PLLLDF |        |    |    |    |    |    |
| W     |    |    | PLLODF |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |    |    |    |    |    |    |    |        | PLLLDF |    |    |    |    |    |
| Reset | X* | X* | X*     | X* | X* | X* | X* | X* | X* | X* | X* | X* | X*     | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X*     | X*     | X* | X* | X* | X* |    |

\* Notes:

- x = Undefined at reset.

## STCU2\_PLL\_CFG field descriptions

| Field           | Description                                                                                                                                                                                                                                                         |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–1<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                             |
| 2–7<br>PLLODF   | PLL Output Division Factor<br><br>The value of this field drives the Output Division factor of the PLL embedded into the device. Refer to the PLL specifications to define the exact mapping between the PLLODF value and the related PLL's Output Division factor. |

Table continues on the next page...



**STCU2\_PLL\_CFG field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                  |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 8–12<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                      |
| 13–15<br>PLLIDF   | PLL Input Division Factor<br>The value of this field drives the Input Division factor of the PLL embedded into the device. Refer to the PLL specifications to define the exact mapping between the PLLIDF value and the related PLL's Input Division factor. |
| 16–24<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                      |
| 25–31<br>PLLLDF   | PLL Loop Division Factor<br>The value of this field drives the Loop Division factor of the PLL embedded into the device. Refer to the PLL specifications to define the exact mapping between the PLLLDF value and the related PLL's Loop Division factor.    |

**56.9.6 STCU2 Watchdog Granularity (STCU2\_WDG)****NOTE**

The reset value is chip-specific; see the chapter that describes how modules are configured and connected.

The STCU2\_WDG register defines the time limit for LBIST and MBIST execution, providing a protection mechanism in case of deadlock or endless conditions during the self test procedure.

When the offline self test sequence is not run and the STCU2\_CFG[BYP] field is not set, bits 15..0 define the time-out before the STCU2\_ERR[WDTO] field is set and the STCU2 core clock is switched off.

In case offline/online self test sequence is run, the STCU2\_WDG value defines the time-out of the execution run.

The register fields are always readable. You can write to these fields when either of the following conditions is true:

- Offline self test phase is active
- Online self test phase is active and CFG[WRP] = 0

Address: 0h base + 14h offset = 14h

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* | 0* |

\* Notes:

•

## STCU2\_WDG field descriptions

| Field          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>WDGEOC | <p>Watchdog End of Count Timer</p> <p>This value defines the time limit for offline and online self test execution to ensure correct operation within the allowed time.</p> <p><b>NOTE:</b> If this register is programmed while STCU2_RUN[RUN]=1, the new STCU2_WDG value, programmed after STCU2_RUN, does not take effect.</p> <p>The allowed values are:</p> <p>0000 0000h: 1 × 16 STCU2 Core clock cycles</p> <p>0000 0001h: 2 × 16 STCU2 Core clock cycles</p> <p>0000 0002h: 3 × 16 STCU2 Core clock cycles</p> <p>...</p> <p>FFFF FFFDh: 4294967294 × 16 STCU2 Core clock cycles</p> <p>FFFF FFFEh: 4294967295 × 16 STCU2 Core clock cycles</p> <p>FFFF FFFFh: 4294967296 × 16 STCU2 Core clock cycles</p> |

### 56.9.7 STCU2 Interrupt Flag Register (STCU2\_INT\_FLG)

The STCU2\_INT\_FLG register includes an MBIST interrupt pending bit. The bit is effective only during the On-Line Self-Test phase and is managed only when the related control bit in the STCU2\_RUNSW register (MBIE) is enabled.

Address: 0h base + 18h offset = 18h

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |        |   |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--|--|--|--|--|--|--|--|--|--|--|--|--|--------|---|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |  |  |  |  |  |  |  |  |  |  |  |  |  |        |   |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |        |   |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |        |   |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |  |  |  |  |  |  |  |  |  |  |  |  |  |        |   |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |  |  |  |  |  |  |  |  |  |  |  |  |  |        |   |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  | MBIFLG | 0 |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  | w1c    |   |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |  |  |  |  |  |  |  |  |  |  |  |  |  |        |   |

**STCU2\_INT\_FLG field descriptions**

| Field            | Description                                                                                                                                                                                        |
|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–29<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                            |
| 30<br>MBIFLG     | MBIST Interrupt Flag<br>0 No interrupt is pending<br>1 An interrupt, highlighting that the On-Line execution of the concurrent scheduled MBIST is completed, is pending. Write 1 to clear the bit. |
| 31<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                            |

56.9.8 STCU2 Error Register (STCU2\_ERR\_STAT)

The STCU2\_ERR\_STAT register includes the status flags related to the STCU2 internal fault conditions occurred during the configuration or the On/Off-Line self testing execution.

The UFSF and RFSF can be set/cleared using the FCCU dedicated channels.

The access to this register is described in the following figure and as follows:

- If you select the byte write capability to write only the UFSF and RFSF, then there is no restriction in writing these bits.
- If your software performs the write operations on other bits besides UFSF/RFSF, then a transfer error is generated only if the value you are writing to the other bits differs from their value currently stored in the register. This functionality has been implemented to prevent potential compilation behavior that might invalidate the UFSF/RFSF single bit set/clean capability.

Address: 0h base + 24h offset = 24h

| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11      | 12     | 13 | 14     | 15     |
|-------|---|---|---|---|---|---|---|---|---|---|----|---------|--------|----|--------|--------|
| R     | 0 |   |   |   |   |   | 0 | 0 |   | 0 |    | LOCKESW | WDTOSW | 0  | ENGESW | INVPSW |
| W     |   |   |   |   |   |   |   |   |   |   |    |         |        |    |        |        |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0       | 0      | 0  | 0      | 0      |

| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22   | 23   | 24 | 25 | 26 | 27    | 28   | 29 | 30   | 31   |
|-------|----|----|----|----|----|----|------|------|----|----|----|-------|------|----|------|------|
| R     | 0  |    |    |    |    |    | UFSF | RFSF | 0  |    |    | LOCKE | WDTO | 0  | ENGE | INVP |
| W     |    |    |    |    |    |    |      |      |    |    |    |       |      |    |      |      |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0    | 0  | 0  | 0  | 0     | 0    | 0  | 0    | 0    |

STCU2\_ERR\_STAT field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–5<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 6<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 7<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 8–10<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 11<br>LOCKESW    | On-Line LOCK Error<br>You can always read this field. The content of this field is initialized to its reset value as soon as STCU2_RUNSW[RUNSW] = 1.<br><br>0 In case PLL is enabled, it is correctly locked during the Self-Test sequence<br>1 When the PLL is enabled, this flag highlight that there has been an unexpected PLL unlock event during the On-Line Self-Test sequence execution. The On-Line Self-test run is stopped and the status of the current running LBISTs or MBISTs is saved into the related registers. The LOCK signal is monitored during the LBIST run when STCU2_RUNSW.LBSWPLEN is set and/or during the MBIST run when STCU2_RUNSW[MBSWPLEN] = 1.                    |
| 12<br>WDTOSW     | On-Line Watchdog time-out<br>You can always read this field. The content of this field is initialized to its reset value as soon as STCU2_RUNSW[RUNSW] = 1.<br><br>0 LBIST and MBIST time slot have been completed within the assigned watchdog time.<br>1 LBIST and MBIST time slot haven't been completed within the assigned watchdog time or there are internal mismatches among End of Execution signals. The conditions that flag the failures are the following: <ul style="list-style-type: none"> <li>• MBIST BEND status flags that at least one of the selected MBIST run is not finished</li> <li>• lbist_done of the selected LBIST flags the LBIST run is/are not finished</li> </ul> |
| 13<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 14<br>ENGESW     | On-Line Engine Error                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

Table continues on the next page...

**STCU2\_ERR\_STAT field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | <p>You can always read this field. The content of this field is initialized to its reset value as soon as STCU2_RUNSW[RUNSW] = 1.</p> <p>0 Valid Engine execution</p> <p>1 Invalid Engine execution. The following conditions set this bit:</p> <ul style="list-style-type: none"> <li>• STCU2 state machines (Watchdog, Master and Loader/Shifter) selects an unused code</li> <li>• Serial pass/fail status of the selected MBIST and the global BBAD flag are not aligned</li> <li>• Serial bend status of the selected MBIST and the global BEND flag are not aligned</li> </ul> |
| 15<br>INVPSW      | <p>On-Line Invalid Pointer</p> <p>You can always read this field. The content of this field is initialized to its reset value as soon as STCU2_RUNSW[RUNSW] = 1.</p> <p>0 Valid linked pointer list</p> <p>1 Invalid linked pointer list. The following conditions set this bit:</p> <ul style="list-style-type: none"> <li>• Initial LBIST or MBIST pointer is out of range</li> <li>• LBIST is selected when MBIST is concurrently running or vice versa</li> <li>• Error in the LBIST/MBIST linking (execution generates an infinite loop)</li> </ul>                             |
| 16–21<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 22<br>UFSF        | <p>Unrecoverable Faults Status Flag</p> <p>This flag reports the global status of the UF. This field can be set or cleared using the FCCU dedicated channel, and can also be set or cleared by software.</p> <p>0 No errors that trigger the Unrecoverable Faults condition</p> <p>1 There are errors that trigger the Unrecoverable Faults condition</p>                                                                                                                                                                                                                            |
| 23<br>RFSF        | <p>Recoverable Faults Status Flag</p> <p>This flag reports the global status of the RF. This field can be set or cleared using the FCCU dedicated channel, and can also be set or cleared by software.</p> <p>0 No errors that trigger the Recoverable Faults condition</p> <p>1 There are errors that trigger the Recoverable Faults condition</p>                                                                                                                                                                                                                                  |
| 24–26<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 27<br>LOCKE       | <p>Off-Line LOCK Error</p> <p>0 The PLL is correctly locked during the Self-Test sequence</p> <p>1 When the PLL is enabled, this flag highlights that there has been an unexpected PLL unlock event during the Off-Line Self-Test sequence execution. The Off-Line Self-test run is stopped and the status of the current running LBISTs or MBISTs is saved into the related registers. The LOCK signal is monitored during the LBIST run when STCU2_RUN[LBPLEN] = 1 and/or during the MBIST run when STCU2_RUN[MBPLEN] = 1.</p>                                                     |
| 28<br>WDTO        | <p>Off-Line Watchdog time-out</p> <p>0 LBIST and MBIST time slot have been completed within the assigned watchdog time.</p> <p>1 LBIST and MBIST time slot haven't been completed within the assigned watchdog time or there are internal mismatches among End of Execution signals. The conditions that flag the failures are the following:</p> <ul style="list-style-type: none"> <li>• The STCU2_RUN[RUN] bit or the STCU2_RUN[BYP] bit is not set before the Watchdog reaches the End of Count</li> </ul>                                                                       |

*Table continues on the next page...*

**STCU2\_ERR\_STAT field descriptions (continued)**

| Field          | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                | <ul style="list-style-type: none"> <li>• MBIST BEND status flags that at least one of the selected MBIST run is not finished</li> <li>• lbist_done of the selected LBIST flags the LBIST run is/are not finished</li> </ul>                                                                                                                                                                                                                                  |
| 29<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                      |
| 30<br>ENGE     | Off-Line Engine Error<br>0 Valid Engine execution<br>1 Invalid Engine execution. The following conditions set this bit: <ul style="list-style-type: none"> <li>• STCU2 state machines (Watchdog, Master and Loader/Shifter) selects an unused code</li> <li>• Serial pass/fail status of the selected MBIST and the global BBAD flag are not aligned</li> <li>• Serial bend status of the selected MBIST and the global BEND flag are not aligned</li> </ul> |
| 31<br>INVP     | Off-Line Invalid pointer<br>0 Valid linked pointer list<br>1 Invalid linked pointer list. The following conditions set this bit: <ul style="list-style-type: none"> <li>• Initial LBIST or MBIST pointer is out of range</li> <li>• LBIST is selected when MBIST is concurrently running or vice versa</li> <li>• Error in the LBIST/MBIST linking (execution generates an infinite loop)</li> </ul>                                                         |

**56.9.9 STCU2 Error FM Register (STCU2\_ERR\_FM)****NOTE**

The reset value is chip-specific; see the chapter that describes how modules are configured and connected.

The STCU2\_ERR\_FM register defines the fault mapping of the STCU2 faults described into the register STCU2\_ERR\_STAT in terms of Unrecoverable or Recoverable Fault. All sources of internal faults can be routed to UF and RF.

The R/W fields in this register are readable at any time. You can write to these fields when either of the following conditions is true:

- Off-line Self-Test phase is active
- On-line Self-Test phase is active and STCU2\_CFG[WRP] = 0

Address: 0h base + 28h offset = 28h

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |

## Register description

| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27       | 28      | 29 | 30      | 31      |
|-------|----|----|----|----|----|----|----|----|----|----|----|----------|---------|----|---------|---------|
| R     | 0  |    |    |    |    |    |    |    |    |    |    | LOCKEUFM | WDTOUFM | 0  | ENGEUFM | INVPUFM |
| W     |    |    |    |    |    |    |    |    |    |    |    |          |         |    |         |         |
| Reset | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x*       | x*      | x* | x*      | x*      |

\* Notes:

- x = Undefined at reset.

## STCU2\_ERR\_FM field descriptions

| Field            | Description                                                                                                   |
|------------------|---------------------------------------------------------------------------------------------------------------|
| 0–26<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                       |
| 27<br>LOCKEUFM   | PLL LOCK Unrecoverable Fault Mapping<br>0 Recoverable Fault Mapping<br>1 Unrecoverable Fault Mapping          |
| 28<br>WDTOUFM    | Watchdog time-out Unrecoverable Fault Mapping<br>0 Recoverable Fault Mapping<br>1 Unrecoverable Fault Mapping |
| 29<br>Reserved   | This field is reserved.<br>This read-only field is reserved and always has the value 0.                       |
| 30<br>ENGEUFM    | Engine Error Unrecoverable fault Mapping<br>0 Recoverable Fault Mapping<br>1 Unrecoverable Fault Mapping      |
| 31<br>INVPUFM    | Invalid Pointer Unrecoverable Fault Mapping<br>0 Recoverable Fault Mapping<br>1 Unrecoverable Mapping         |

## 56.9.10 STCU2 Off-Line LBIST Status Register (STCU2\_LBS)

The STCU2\_LBS register includes the results corresponding to the execution of the selected Off-Line LBIST.

The size of the register depends on the number of LBIST .

Address: 0h base + 2Ch offset = 2Ch

| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |



|       |    |    |    |    |    |    |    |    |    |    |      |      |      |      |      |      |
|-------|----|----|----|----|----|----|----|----|----|----|------|------|------|------|------|------|
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26   | 27   | 28   | 29   | 30   | 31   |
| R     | 0  |    |    |    |    |    |    |    |    |    | LBS5 | LBS4 | LBS3 | LBS2 | LBS1 | LBS0 |
| W     |    |    |    |    |    |    |    |    |    |    |      |      |      |      |      |      |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0    | 0    | 0    | 0    | 0    |

**STCU2\_LBS field descriptions**

| Field            | Description                                                                                       |
|------------------|---------------------------------------------------------------------------------------------------|
| 0–25<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.           |
| 26<br>LBS5       | Off-Line status of the selected LBIST<br>0 Failed LBIST execution<br>1 Successful LBIST execution |
| 27<br>LBS4       | Off-Line status of the selected LBIST<br>0 Failed LBIST execution<br>1 Successful LBIST execution |
| 28<br>LBS3       | Off-Line status of the selected LBIST<br>0 Failed LBIST execution<br>1 Successful LBIST execution |
| 29<br>LBS2       | Off-Line status of the selected LBIST<br>0 Failed LBIST execution<br>1 Successful LBIST execution |
| 30<br>LBS1       | Off-Line status of the selected LBIST<br>0 Failed LBIST execution<br>1 Successful LBIST execution |
| 31<br>LBS0       | Off-Line status of the selected LBIST<br>0 Failed LBIST execution<br>1 Successful LBIST execution |

**56.9.11 STCU2 Off-Line LBIST End Flag Register (STCU2\_LBE)**

The STCU2\_LBE register includes the End Flag related to the execution of the selected Off-Line LBIST.

The size of the register depends on the number of LBIST .

Address: 0h base + 30h offset = 30h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

## Register description

| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26   | 27   | 28   | 29   | 30   | 31   |
|-------|----|----|----|----|----|----|----|----|----|----|------|------|------|------|------|------|
| R     | 0  |    |    |    |    |    |    |    |    |    | LBE5 | LBE4 | LBE3 | LBE2 | LBE1 | LBE0 |
| W     |    |    |    |    |    |    |    |    |    |    |      |      |      |      |      |      |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0    | 0    | 0    | 0    | 0    |

### STCU2\_LBE field descriptions

| Field            | Description                                                                                                    |
|------------------|----------------------------------------------------------------------------------------------------------------|
| 0–25<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                        |
| 26<br>LBE5       | Off-Line End status of the selected LBIST<br>0 LBIST execution not yet completed<br>1 LBIST execution finished |
| 27<br>LBE4       | Off-Line End status of the selected LBIST<br>0 LBIST execution not yet completed<br>1 LBIST execution finished |
| 28<br>LBE3       | Off-Line End status of the selected LBIST<br>0 LBIST execution not yet completed<br>1 LBIST execution finished |
| 29<br>LBE2       | Off-Line End status of the selected LBIST<br>0 LBIST execution not yet completed<br>1 LBIST execution finished |
| 30<br>LBE1       | Off-Line End status of the selected LBIST<br>0 LBIST execution not yet completed<br>1 LBIST execution finished |
| 31<br>LBE0       | Off-Line End status of the selected LBIST<br>0 LBIST execution not yet completed<br>1 LBIST execution finished |

## 56.9.12 STCU2 On-Line LBIST Status Register (STCU2\_LBSSW)

The STCU2\_LBSSW register includes the results corresponding to the execution of the selected On-Line LBIST.

The size of the register depends on the number of LBIST .

The content of this register is initialized to its reset value as soon as STCU2\_RUNSW[RUNSW] = 1.

Address: 0h base + 34h offset = 34h

|       |    |    |    |    |    |    |    |    |    |    |        |        |        |        |        |        |
|-------|----|----|----|----|----|----|----|----|----|----|--------|--------|--------|--------|--------|--------|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10     | 11     | 12     | 13     | 14     | 15     |
| R     | 0  |    |    |    |    |    |    |    |    |    |        |        |        |        |        |        |
| W     |    |    |    |    |    |    |    |    |    |    |        |        |        |        |        |        |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0      | 0      | 0      | 0      | 0      | 0      |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26     | 27     | 28     | 29     | 30     | 31     |
| R     | 0  |    |    |    |    |    |    |    |    |    | LBSSW5 | LBSSW4 | LBSSW3 | LBSSW2 | LBSSW1 | LBSSW0 |
| W     |    |    |    |    |    |    |    |    |    |    |        |        |        |        |        |        |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0      | 0      | 0      | 0      | 0      | 0      |

**STCU2\_LBSSW field descriptions**

| Field            | Description                                                                                              |
|------------------|----------------------------------------------------------------------------------------------------------|
| 0–25<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                  |
| 26<br>LBSSW5     | LBSSWx: On-Line status of the selected LBIST<br>0 Failed LBIST execution<br>1 Successful LBIST execution |
| 27<br>LBSSW4     | LBSSWx: On-Line status of the selected LBIST<br>0 Failed LBIST execution<br>1 Successful LBIST execution |
| 28<br>LBSSW3     | LBSSWx: On-Line status of the selected LBIST<br>0 Failed LBIST execution<br>1 Successful LBIST execution |
| 29<br>LBSSW2     | LBSSWx: On-Line status of the selected LBIST<br>0 Failed LBIST execution<br>1 Successful LBIST execution |
| 30<br>LBSSW1     | LBSSWx: On-Line status of the selected LBIST                                                             |

*Table continues on the next page...*

**STCU2\_LBSSW field descriptions (continued)**

| Field        | Description                                                                                              |
|--------------|----------------------------------------------------------------------------------------------------------|
|              | 0 Failed LBIST execution<br>1 Successful LBIST execution                                                 |
| 31<br>LBSSW0 | LBSSWx: On-Line status of the selected LBIST<br>0 Failed LBIST execution<br>1 Successful LBIST execution |

**56.9.13 STCU2 On-Line LBIST End Flag Register (STCU2\_LBESW)**

The STCU2\_LBESW register includes the End Flag related to the execution of the selected On-Line LBIST.

The size of the register depends on the number of LBIST.

The content of this register is initialized to its reset value as soon as STCU2\_RUNSW[RUNSW] = 1.

Address: 0h base + 38h offset = 38h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |    |    |    |    |    |    |    |    |    |        |    |        |    |        |    |        |  |        |  |        |  |
|-------|----|----|----|----|----|----|----|----|----|----|--------|----|--------|----|--------|----|--------|--|--------|--|--------|--|
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26     | 27 | 28     | 29 | 30     | 31 |        |  |        |  |        |  |
| R     | 0  |    |    |    |    |    |    |    |    |    | LBESW5 |    | LBESW4 |    | LBESW3 |    | LBESW2 |  | LBESW1 |  | LBESW0 |  |
| W     |    |    |    |    |    |    |    |    |    |    |        |    |        |    |        |    |        |  |        |  |        |  |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0      | 0  | 0      | 0  | 0      | 0  |        |  |        |  |        |  |

**STCU2\_LBESW field descriptions**

| Field            | Description                                                                                           |
|------------------|-------------------------------------------------------------------------------------------------------|
| 0–25<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.               |
| 26<br>LBESW5     | LBESWx: On-Line LBIST End status<br>0 LBIST execution not yet completed<br>1 LBIST execution finished |
| 27<br>LBESW4     | LBESWx: On-Line LBIST End status<br>0 LBIST execution not yet completed<br>1 LBIST execution finished |
| 28<br>LBESW3     | LBESWx: On-Line LBIST End status<br>0 LBIST execution not yet completed<br>1 LBIST execution finished |
| 29<br>LBESW2     | LBESWx: On-Line LBIST End status<br>0 LBIST execution not yet completed<br>1 LBIST execution finished |
| 30<br>LBESW1     | LBESWx: On-Line LBIST End status<br>0 LBIST execution not yet completed<br>1 LBIST execution finished |
| 31<br>LBESW0     | LBESWx: On-Line LBIST End status<br>0 LBIST execution not yet completed<br>1 LBIST execution finished |

**56.9.14 STCU2 On-Line LBIST Reset Management (STCU2\_LBRMSW)****NOTE**

The reset value is chip-specific; see the chapter that describes how modules are configured and connected.

The STCU2\_LBRMSW register defines the On-Line reset mapping for each LBIST. Global or specific functional reset can be programmed depending on the On-Line Self-Test requirements and top level integration architecture .

The size of the register depends on the number of LBIST .

The R/W fields in this register are readable at any time. You can write to these fields when either of the following conditions is true:

- Off-line Self-Test phase is active
- On-line Self-Test phase is active and STCU2\_CFG[WRP] = 0

## Register description

Address: 0h base + 3Ch offset = 3Ch

|       |    |    |    |    |    |    |    |    |    |    |         |         |         |         |         |         |
|-------|----|----|----|----|----|----|----|----|----|----|---------|---------|---------|---------|---------|---------|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10      | 11      | 12      | 13      | 14      | 15      |
| R     | 0  |    |    |    |    |    |    |    |    |    |         |         |         |         |         |         |
| W     |    |    |    |    |    |    |    |    |    |    |         |         |         |         |         |         |
| Reset | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x*      | x*      | x*      | x*      | x*      | x*      |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26      | 27      | 28      | 29      | 30      | 31      |
| R     | 0  |    |    |    |    |    |    |    |    |    | LBRMSW5 | LBRMSW4 | LBRMSW3 | LBRMSW2 | LBRMSW1 | LBRMSW0 |
| W     |    |    |    |    |    |    |    |    |    |    |         |         |         |         |         |         |
| Reset | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x*      | x*      | x*      | x*      | x*      | x*      |

\* Notes:

- x = Undefined at reset.

## STCU2\_LBRMSW field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                          |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–25<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                              |
| 26<br>LBRMSW5    | On-Line LBIST Reset Management<br><b>NOTE:</b> In case one of the selected LBIST has this bit set to '1', then only the Global functional reset will be pulsed.<br><br>0 Dedicated functional reset is pulsed at the end of LBIST run<br>1 Global functional reset is pulsed at the end of LBIST run |
| 27<br>LBRMSW4    | On-Line LBIST Reset Management<br><b>NOTE:</b> In case one of the selected LBIST has this bit set to '1', then only the Global functional reset will be pulsed.<br><br>0 Dedicated functional reset is pulsed at the end of LBIST run<br>1 Global functional reset is pulsed at the end of LBIST run |
| 28<br>LBRMSW3    | On-Line LBIST Reset Management<br><b>NOTE:</b> In case one of the selected LBIST has this bit set to '1', then only the Global functional reset will be pulsed.<br><br>0 Dedicated functional reset is pulsed at the end of LBIST run<br>1 Global functional reset is pulsed at the end of LBIST run |
| 29<br>LBRMSW2    | On-Line LBIST Reset Management<br><b>NOTE:</b> In case one of the selected LBIST has this bit set to '1', then only the Global functional reset will be pulsed.<br><br>0 Dedicated functional reset is pulsed at the end of LBIST run<br>1 Global functional reset is pulsed at the end of LBIST run |
| 30<br>LBRMSW1    | On-Line LBIST Reset Management                                                                                                                                                                                                                                                                       |

Table continues on the next page...

**STCU2\_LBRMSW field descriptions (continued)**

| Field         | Description                                                                                                                                                                                                                                                                                              |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               | <b>NOTE:</b> In case one of the selected LBIST has this bit set to '1', then only the Global functional reset will be pulsed.<br><br>0 Dedicated functional reset is pulsed at the end of LBIST run<br>1 Global functional reset is pulsed at the end of LBIST run                                       |
| 31<br>LBRMSW0 | On-Line LBIST Reset Management<br><br><b>NOTE:</b> In case one of the selected LBIST has this bit set to '1', then only the Global functional reset will be pulsed.<br><br>0 Dedicated functional reset is pulsed at the end of LBIST run<br>1 Global functional reset is pulsed at the end of LBIST run |

**56.9.15 STCU2 LBIST Unrecoverable FM Register (STCU2\_LBUFM)****NOTE**

The reset value is chip-specific; see the chapter that describes how modules are configured and connected.

The STCU2\_LBUFM register defines the fault mapping of each LBIST in terms of Unrecoverable or Recoverable Fault.

The size of the register depends on the number of LBIST .

The R/W fields in this register are readable at any time. You can write to these fields when either of the following conditions is true:

- Off-line Self-Test phase is active
- On-line Self-Test phase is active and STCU2\_CFG[WRP] = 0

Address: 0h base + 40h offset = 40h

|       |    |    |    |    |    |    |    |    |    |    |        |        |        |        |        |        |
|-------|----|----|----|----|----|----|----|----|----|----|--------|--------|--------|--------|--------|--------|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10     | 11     | 12     | 13     | 14     | 15     |
| R     | 0  |    |    |    |    |    |    |    |    |    |        |        |        |        |        |        |
| W     |    |    |    |    |    |    |    |    |    |    |        |        |        |        |        |        |
| Reset | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x*     | x*     | x*     | x*     | x*     | x*     |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26     | 27     | 28     | 29     | 30     | 31     |
| R     | 0  |    |    |    |    |    |    |    |    |    | LBUFM5 | LBUFM4 | LBUFM3 | LBUFM2 | LBUFM1 | LBUFM0 |
| W     |    |    |    |    |    |    |    |    |    |    |        |        |        |        |        |        |
| Reset | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x*     | x*     | x*     | x*     | x*     | x*     |

\* Notes:

## Register description

- x = Undefined at reset.

### STCU2\_LBUFM field descriptions

| Field            | Description                                                                                       |
|------------------|---------------------------------------------------------------------------------------------------|
| 0–25<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.           |
| 26<br>LBUFM5     | LBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 27<br>LBUFM4     | LBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 28<br>LBUFM3     | LBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 29<br>LBUFM2     | LBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 30<br>LBUFM1     | LBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 31<br>LBUFM0     | LBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |

## 56.9.16 STCU2 Off-Line MBIST Status Low Register (STCU2\_MBSL)

The STCU2\_MBSL register includes the results corresponding to the execution of the selected Off-Line MBIST in the range NMCUT = 0-31.

The size of the register depends on the number of BISTed RAMs/ROMs .



Address: 0h base + 44h offset = 44h

|       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit   | 0     | 1     | 2     | 3     | 4     | 5     | 6     | 7     | 8     | 9     | 10    | 11    | 12    | 13    | 14    | 15    |
| R     | MBS31 | MBS30 | MBS29 | MBS28 | MBS27 | MBS26 | MBS25 | MBS24 | MBS23 | MBS22 | MBS21 | MBS20 | MBS19 | MBS18 | MBS17 | MBS16 |
| W     |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

|       |       |       |       |       |       |       |      |      |      |      |      |      |      |      |      |      |
|-------|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| Bit   | 16    | 17    | 18    | 19    | 20    | 21    | 22   | 23   | 24   | 25   | 26   | 27   | 28   | 29   | 30   | 31   |
| R     | MBS15 | MBS14 | MBS13 | MBS12 | MBS11 | MBS10 | MBS9 | MBS8 | MBS7 | MBS6 | MBS5 | MBS4 | MBS3 | MBS2 | MBS1 | MBS0 |
| W     |       |       |       |       |       |       |      |      |      |      |      |      |      |      |      |      |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |

**STCU2\_MBSL field descriptions**

| Field      | Description                                                                                                                                                                                                                                                                         |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>MBS31 | Off-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><br><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEL register reports the MBIST is finished.<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 1<br>MBS30 | Off-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><br><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEL register reports the MBIST is finished.<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 2<br>MBS29 | Off-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><br><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEL register reports the MBIST is finished.<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |

*Table continues on the next page...*

## STCU2\_MBSL field descriptions (continued)

| Field       | Description                                                                                                                                                                                                                                                                         |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3<br>MBS28  | Off-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><br><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEL register reports the MBIST is finished.<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 4<br>MBS27  | Off-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><br><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEL register reports the MBIST is finished.<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 5<br>MBS26  | Off-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><br><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEL register reports the MBIST is finished.<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 6<br>MBS25  | Off-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><br><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEL register reports the MBIST is finished.<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 7<br>MBS24  | Off-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><br><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEL register reports the MBIST is finished.<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 8<br>MBS23  | Off-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><br><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEL register reports the MBIST is finished.<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 9<br>MBS22  | Off-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><br><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEL register reports the MBIST is finished.<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 10<br>MBS21 | Off-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><br><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEL register reports the MBIST is finished.                                                                                             |

*Table continues on the next page...*

## STCU2\_MBSL field descriptions (continued)

| Field       | Description                                                                                                                                                                                                                                                                 |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|             | 0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution                                                                                                                                                                                        |
| 11<br>MBS20 | Off-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEL register reports the MBIST is finished.<br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 12<br>MBS19 | Off-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEL register reports the MBIST is finished.<br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 13<br>MBS18 | Off-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEL register reports the MBIST is finished.<br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 14<br>MBS17 | Off-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEL register reports the MBIST is finished.<br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 15<br>MBS16 | Off-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEL register reports the MBIST is finished.<br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 16<br>MBS15 | Off-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEL register reports the MBIST is finished.<br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 17<br>MBS14 | Off-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEL register reports the MBIST is finished.<br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 18<br>MBS13 | Off-Line status (NMCUT range = 0 to 31) of the selected MBIST                                                                                                                                                                                                               |

*Table continues on the next page...*

**STCU2\_MBSL field descriptions (continued)**

| Field       | Description                                                                                                                                                                                                                                                                                 |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|             | <p><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEL register reports the MBIST is finished.</p> <p>0 Failed NMCUT BIST execution<br/>1 No Fault detected during the NMCUT BIST execution</p>                                                                      |
| 19<br>MBS12 | <p>Off-Line status (NMCUT range = 0 to 31) of the selected MBIST</p> <p><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEL register reports the MBIST is finished.</p> <p>0 Failed NMCUT BIST execution<br/>1 No Fault detected during the NMCUT BIST execution</p> |
| 20<br>MBS11 | <p>Off-Line status (NMCUT range = 0 to 31) of the selected MBIST</p> <p><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEL register reports the MBIST is finished.</p> <p>0 Failed NMCUT BIST execution<br/>1 No Fault detected during the NMCUT BIST execution</p> |
| 21<br>MBS10 | <p>Off-Line status (NMCUT range = 0 to 31) of the selected MBIST</p> <p><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEL register reports the MBIST is finished.</p> <p>0 Failed NMCUT BIST execution<br/>1 No Fault detected during the NMCUT BIST execution</p> |
| 22<br>MBS9  | <p>Off-Line status (NMCUT range = 0 to 31) of the selected MBIST</p> <p><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEL register reports the MBIST is finished.</p> <p>0 Failed NMCUT BIST execution<br/>1 No Fault detected during the NMCUT BIST execution</p> |
| 23<br>MBS8  | <p>Off-Line status (NMCUT range = 0 to 31) of the selected MBIST</p> <p><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEL register reports the MBIST is finished.</p> <p>0 Failed NMCUT BIST execution<br/>1 No Fault detected during the NMCUT BIST execution</p> |
| 24<br>MBS7  | <p>Off-Line status (NMCUT range = 0 to 31) of the selected MBIST</p> <p><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEL register reports the MBIST is finished.</p> <p>0 Failed NMCUT BIST execution<br/>1 No Fault detected during the NMCUT BIST execution</p> |
| 25<br>MBS6  | <p>Off-Line status (NMCUT range = 0 to 31) of the selected MBIST</p> <p><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEL register reports the MBIST is finished.</p> <p>0 Failed NMCUT BIST execution<br/>1 No Fault detected during the NMCUT BIST execution</p> |

*Table continues on the next page...*

**STCU2\_MBSL field descriptions (continued)**

| Field      | Description                                                                                                                                                                                                                                                                         |
|------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 26<br>MBS5 | Off-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><br><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEL register reports the MBIST is finished.<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 27<br>MBS4 | Off-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><br><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEL register reports the MBIST is finished.<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 28<br>MBS3 | Off-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><br><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEL register reports the MBIST is finished.<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 29<br>MBS2 | Off-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><br><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEL register reports the MBIST is finished.<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 30<br>MBS1 | Off-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><br><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEL register reports the MBIST is finished.<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 31<br>MBS0 | Off-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><br><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEL register reports the MBIST is finished.<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |

**56.9.17 STCU2 Off-Line MBIST Status Medium Register (STCU2\_MBSM)**

The STCU2\_MBSM register includes the results corresponding to the execution of the selected Off-Line MBIST in the range NMCUT = 32-63.

The size of the register depends on the number of BISTed RAMs/ROMs .

## Register description

Address: 0h base + 48h offset = 48h

|       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit   | 0     | 1     | 2     | 3     | 4     | 5     | 6     | 7     | 8     | 9     | 10    | 11    | 12    | 13    | 14    | 15    |
| R     | MBS63 | MBS62 | MBS61 | MBS60 | MBS59 | MBS58 | MBS57 | MBS56 | MBS55 | MBS54 | MBS53 | MBS52 | MBS51 | MBS50 | MBS49 | MBS48 |
| W     |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

|       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit   | 16    | 17    | 18    | 19    | 20    | 21    | 22    | 23    | 24    | 25    | 26    | 27    | 28    | 29    | 30    | 31    |
| R     | MBS47 | MBS46 | MBS45 | MBS44 | MBS43 | MBS42 | MBS41 | MBS40 | MBS39 | MBS38 | MBS37 | MBS36 | MBS35 | MBS34 | MBS33 | MBS32 |
| W     |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

## STCU2\_MBSM field descriptions

| Field      | Description                                                                                                                                                                                                                                                                                         |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>MBS63 | <p>MBSx: Off-Line status (NMCUT range = 32 to 63) of the selected MBIST</p> <p><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEM register reports the MBIST is finished).</p> <p>0 Failed NMCUT BIST execution<br/>1 No Fault detected during the NMCUT BIST execution</p> |
| 1<br>MBS62 | <p>MBSx: Off-Line status (NMCUT range = 32 to 63) of the selected MBIST</p> <p><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEM register reports the MBIST is finished).</p> <p>0 Failed NMCUT BIST execution<br/>1 No Fault detected during the NMCUT BIST execution</p> |
| 2<br>MBS61 | <p>MBSx: Off-Line status (NMCUT range = 32 to 63) of the selected MBIST</p> <p><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEM register reports the MBIST is finished).</p> <p>0 Failed NMCUT BIST execution<br/>1 No Fault detected during the NMCUT BIST execution</p> |

Table continues on the next page...

**STCU2\_MBSM field descriptions (continued)**

| Field       | Description                                                                                                                                                                                                                                                                                            |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3<br>MBS60  | <p>MBSx: Off-Line status (NMCUT range = 32 to 63) of the selected MBIST</p> <p><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEM register reports the MBIST is finished).</p> <p>0 Failed NMCUT BIST execution</p> <p>1 No Fault detected during the NMCUT BIST execution</p> |
| 4<br>MBS59  | <p>MBSx: Off-Line status (NMCUT range = 32 to 63) of the selected MBIST</p> <p><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEM register reports the MBIST is finished).</p> <p>0 Failed NMCUT BIST execution</p> <p>1 No Fault detected during the NMCUT BIST execution</p> |
| 5<br>MBS58  | <p>MBSx: Off-Line status (NMCUT range = 32 to 63) of the selected MBIST</p> <p><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEM register reports the MBIST is finished).</p> <p>0 Failed NMCUT BIST execution</p> <p>1 No Fault detected during the NMCUT BIST execution</p> |
| 6<br>MBS57  | <p>MBSx: Off-Line status (NMCUT range = 32 to 63) of the selected MBIST</p> <p><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEM register reports the MBIST is finished).</p> <p>0 Failed NMCUT BIST execution</p> <p>1 No Fault detected during the NMCUT BIST execution</p> |
| 7<br>MBS56  | <p>MBSx: Off-Line status (NMCUT range = 32 to 63) of the selected MBIST</p> <p><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEM register reports the MBIST is finished).</p> <p>0 Failed NMCUT BIST execution</p> <p>1 No Fault detected during the NMCUT BIST execution</p> |
| 8<br>MBS55  | <p>MBSx: Off-Line status (NMCUT range = 32 to 63) of the selected MBIST</p> <p><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEM register reports the MBIST is finished).</p> <p>0 Failed NMCUT BIST execution</p> <p>1 No Fault detected during the NMCUT BIST execution</p> |
| 9<br>MBS54  | <p>MBSx: Off-Line status (NMCUT range = 32 to 63) of the selected MBIST</p> <p><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEM register reports the MBIST is finished).</p> <p>0 Failed NMCUT BIST execution</p> <p>1 No Fault detected during the NMCUT BIST execution</p> |
| 10<br>MBS53 | <p>MBSx: Off-Line status (NMCUT range = 32 to 63) of the selected MBIST</p> <p><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEM register reports the MBIST is finished).</p>                                                                                                 |

*Table continues on the next page...*

## STCU2\_MBSM field descriptions (continued)

| Field       | Description                                                                                                                                                                                                                                                                         |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|             | 0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution                                                                                                                                                                                                |
| 11<br>MBS52 | MBSx: Off-Line status (NMCUT range = 32 to 63) of the selected MBIST<br><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEM register reports the MBIST is finished).<br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 12<br>MBS51 | MBSx: Off-Line status (NMCUT range = 32 to 63) of the selected MBIST<br><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEM register reports the MBIST is finished).<br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 13<br>MBS50 | MBSx: Off-Line status (NMCUT range = 32 to 63) of the selected MBIST<br><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEM register reports the MBIST is finished).<br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 14<br>MBS49 | MBSx: Off-Line status (NMCUT range = 32 to 63) of the selected MBIST<br><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEM register reports the MBIST is finished).<br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 15<br>MBS48 | MBSx: Off-Line status (NMCUT range = 32 to 63) of the selected MBIST<br><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEM register reports the MBIST is finished).<br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 16<br>MBS47 | MBSx: Off-Line status (NMCUT range = 32 to 63) of the selected MBIST<br><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEM register reports the MBIST is finished).<br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 17<br>MBS46 | MBSx: Off-Line status (NMCUT range = 32 to 63) of the selected MBIST<br><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEM register reports the MBIST is finished).<br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 18<br>MBS45 | MBSx: Off-Line status (NMCUT range = 32 to 63) of the selected MBIST                                                                                                                                                                                                                |

Table continues on the next page...



**STCU2\_MBSM field descriptions (continued)**

| Field       | Description                                                                                                                                                                                                                                                                                            |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|             | <p><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEM register reports the MBIST is finished).</p> <p>0 Failed NMCUT BIST execution</p> <p>1 No Fault detected during the NMCUT BIST execution</p>                                                                             |
| 19<br>MBS44 | <p>MBSx: Off-Line status (NMCUT range = 32 to 63) of the selected MBIST</p> <p><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEM register reports the MBIST is finished).</p> <p>0 Failed NMCUT BIST execution</p> <p>1 No Fault detected during the NMCUT BIST execution</p> |
| 20<br>MBS43 | <p>MBSx: Off-Line status (NMCUT range = 32 to 63) of the selected MBIST</p> <p><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEM register reports the MBIST is finished).</p> <p>0 Failed NMCUT BIST execution</p> <p>1 No Fault detected during the NMCUT BIST execution</p> |
| 21<br>MBS42 | <p>MBSx: Off-Line status (NMCUT range = 32 to 63) of the selected MBIST</p> <p><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEM register reports the MBIST is finished).</p> <p>0 Failed NMCUT BIST execution</p> <p>1 No Fault detected during the NMCUT BIST execution</p> |
| 22<br>MBS41 | <p>MBSx: Off-Line status (NMCUT range = 32 to 63) of the selected MBIST</p> <p><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEM register reports the MBIST is finished).</p> <p>0 Failed NMCUT BIST execution</p> <p>1 No Fault detected during the NMCUT BIST execution</p> |
| 23<br>MBS40 | <p>MBSx: Off-Line status (NMCUT range = 32 to 63) of the selected MBIST</p> <p><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEM register reports the MBIST is finished).</p> <p>0 Failed NMCUT BIST execution</p> <p>1 No Fault detected during the NMCUT BIST execution</p> |
| 24<br>MBS39 | <p>MBSx: Off-Line status (NMCUT range = 32 to 63) of the selected MBIST</p> <p><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEM register reports the MBIST is finished).</p> <p>0 Failed NMCUT BIST execution</p> <p>1 No Fault detected during the NMCUT BIST execution</p> |
| 25<br>MBS38 | <p>MBSx: Off-Line status (NMCUT range = 32 to 63) of the selected MBIST</p> <p><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEM register reports the MBIST is finished).</p> <p>0 Failed NMCUT BIST execution</p> <p>1 No Fault detected during the NMCUT BIST execution</p> |

*Table continues on the next page...*

**STCU2\_MBSM field descriptions (continued)**

| Field       | Description                                                                                                                                                                                                                                                                                         |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 26<br>MBS37 | <p>MBSx: Off-Line status (NMCUT range = 32 to 63) of the selected MBIST</p> <p><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEM register reports the MBIST is finished).</p> <p>0 Failed NMCUT BIST execution<br/>1 No Fault detected during the NMCUT BIST execution</p> |
| 27<br>MBS36 | <p>MBSx: Off-Line status (NMCUT range = 32 to 63) of the selected MBIST</p> <p><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEM register reports the MBIST is finished).</p> <p>0 Failed NMCUT BIST execution<br/>1 No Fault detected during the NMCUT BIST execution</p> |
| 28<br>MBS35 | <p>MBSx: Off-Line status (NMCUT range = 32 to 63) of the selected MBIST</p> <p><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEM register reports the MBIST is finished).</p> <p>0 Failed NMCUT BIST execution<br/>1 No Fault detected during the NMCUT BIST execution</p> |
| 29<br>MBS34 | <p>MBSx: Off-Line status (NMCUT range = 32 to 63) of the selected MBIST</p> <p><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEM register reports the MBIST is finished).</p> <p>0 Failed NMCUT BIST execution<br/>1 No Fault detected during the NMCUT BIST execution</p> |
| 30<br>MBS33 | <p>MBSx: Off-Line status (NMCUT range = 32 to 63) of the selected MBIST</p> <p><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEM register reports the MBIST is finished).</p> <p>0 Failed NMCUT BIST execution<br/>1 No Fault detected during the NMCUT BIST execution</p> |
| 31<br>MBS32 | <p>MBSx: Off-Line status (NMCUT range = 32 to 63) of the selected MBIST</p> <p><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEM register reports the MBIST is finished).</p> <p>0 Failed NMCUT BIST execution<br/>1 No Fault detected during the NMCUT BIST execution</p> |

**56.9.18 STCU2 Off-Line MBIST Status High Register (STCU2\_MBSH)**

The STCU2\_MBSH register includes the results corresponding to the execution of the selected Off-Line MBIST in the range NMCUT = 64-95.

The size of the register depends on the number of BISTed RAMs/ROMs.

Address: 0h base + 4Ch offset = 4Ch

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |       |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--|--|--|--|--|--|--|--|--|--|--|--|--|--|-------|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |       |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |       |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |       |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |       |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |       |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  | MBS64 |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |  |  |  |  |  |  |  |  |  |  |  |  |  |       |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |       |

STCU2\_MBSH field descriptions

| Field            | Description                                                                                                                                                                                                                                                                           |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–30<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                               |
| 31<br>MBS64      | Off-Line status (NMCUT range = 64 to 95) of the selected MBIST<br><br><b>NOTE:</b> This bit is meaningful when the related bit of the STCU2_MBEH register reports the MBIST is finished).<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |

### 56.9.19 STCU2 Off-Line MBIST End Flag Low Register (STCU2\_MBEL)

The STCU2\_MBEL register includes the End Flag related to the execution of the selected Off-Line MBIST in the range NMCUT = 0-31.

The size of the register depends on the number of BISTed RAMs/ROMs.

## Register description

Address: 0h base + 50h offset = 50h

|       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit   | 0     | 1     | 2     | 3     | 4     | 5     | 6     | 7     | 8     | 9     | 10    | 11    | 12    | 13    | 14    | 15    |
| R     | MBE31 | MBE30 | MBE29 | MBE28 | MBE27 | MBE26 | MBE25 | MBE24 | MBE23 | MBE22 | MBE21 | MBE20 | MBE19 | MBE18 | MBE17 | MBE16 |
| W     |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

|       |       |       |       |       |       |       |      |      |      |      |      |      |      |      |      |      |
|-------|-------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|------|
| Bit   | 16    | 17    | 18    | 19    | 20    | 21    | 22   | 23   | 24   | 25   | 26   | 27   | 28   | 29   | 30   | 31   |
| R     | MBE15 | MBE14 | MBE13 | MBE12 | MBE11 | MBE10 | MBE9 | MBE8 | MBE7 | MBE6 | MBE5 | MBE4 | MBE3 | MBE2 | MBE1 | MBE0 |
| W     |       |       |       |       |       |       |      |      |      |      |      |      |      |      |      |      |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    | 0    |

## STCU2\_MBEL field descriptions

| Field      | Description                                                                                                          |
|------------|----------------------------------------------------------------------------------------------------------------------|
| 0<br>MBE31 | Off-Line End status (0 to 31) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 1<br>MBE30 | Off-Line End status (0 to 31) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 2<br>MBE29 | Off-Line End status (0 to 31) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 3<br>MBE28 | Off-Line End status (0 to 31) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 4<br>MBE27 | Off-Line End status (0 to 31) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |

Table continues on the next page...

**STCU2\_MBEL field descriptions (continued)**

| <b>Field</b> | <b>Description</b>                                                                                                   |
|--------------|----------------------------------------------------------------------------------------------------------------------|
| 5<br>MBE26   | Off-Line End status (0 to 31) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 6<br>MBE25   | Off-Line End status (0 to 31) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 7<br>MBE24   | Off-Line End status (0 to 31) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 8<br>MBE23   | Off-Line End status (0 to 31) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 9<br>MBE22   | Off-Line End status (0 to 31) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 10<br>MBE21  | Off-Line End status (0 to 31) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 11<br>MBE20  | Off-Line End status (0 to 31) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 12<br>MBE19  | Off-Line End status (0 to 31) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 13<br>MBE18  | Off-Line End status (0 to 31) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 14<br>MBE17  | Off-Line End status (0 to 31) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 15<br>MBE16  | Off-Line End status (0 to 31) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 16<br>MBE15  | Off-Line End status (0 to 31) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |

*Table continues on the next page...*

**STCU2\_MBEL field descriptions (continued)**

| Field       | Description                                                                                                          |
|-------------|----------------------------------------------------------------------------------------------------------------------|
| 17<br>MBE14 | Off-Line End status (0 to 31) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 18<br>MBE13 | Off-Line End status (0 to 31) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 19<br>MBE12 | Off-Line End status (0 to 31) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 20<br>MBE11 | Off-Line End status (0 to 31) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 21<br>MBE10 | Off-Line End status (0 to 31) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 22<br>MBE9  | Off-Line End status (0 to 31) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 23<br>MBE8  | Off-Line End status (0 to 31) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 24<br>MBE7  | Off-Line End status (0 to 31) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 25<br>MBE6  | Off-Line End status (0 to 31) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 26<br>MBE5  | Off-Line End status (0 to 31) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 27<br>MBE4  | Off-Line End status (0 to 31) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 28<br>MBE3  | Off-Line End status (0 to 31) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |

*Table continues on the next page...*

**STCU2\_MBEL field descriptions (continued)**

| Field      | Description                                                                                                          |
|------------|----------------------------------------------------------------------------------------------------------------------|
| 29<br>MBE2 | Off-Line End status (0 to 31) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 30<br>MBE1 | Off-Line End status (0 to 31) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 31<br>MBE0 | Off-Line End status (0 to 31) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |

**56.9.20 STCU2 Off-Line MBIST End Flag Medium Register (STCU2\_MBEM)**

The STCU2\_MBEM register includes the End Flag related to the execution of the selected Off-Line MBIST in the range NMCUT = 32-63.

The size of the register depends on the number of BISTed RAMs/ROMs.

Address: 0h base + 54h offset = 54h

|       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Bit   | 0     | 1     | 2     | 3     | 4     | 5     | 6     | 7     | 8     | 9     | 10    | 11    | 12    | 13    | 14    | 15    |
| R     | MBE63 | MBE62 | MBE61 | MBE60 | MBE59 | MBE58 | MBE57 | MBE56 | MBE55 | MBE54 | MBE53 | MBE52 | MBE51 | MBE50 | MBE49 | MBE48 |
| W     |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

## Register description

| Bit   | 16    | 17    | 18    | 19    | 20    | 21    | 22    | 23    | 24    | 25    | 26    | 27    | 28    | 29    | 30    | 31    |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| R     | MBE47 | MBE46 | MBE45 | MBE44 | MBE43 | MBE42 | MBE41 | MBE40 | MBE39 | MBE38 | MBE37 | MBE36 | MBE35 | MBE34 | MBE33 | MBE32 |
| W     |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
| Reset | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     | 0     |

### STCU2\_MBEM field descriptions

| Field      | Description                                                                                                           |
|------------|-----------------------------------------------------------------------------------------------------------------------|
| 0<br>MBE63 | Off-Line End status (32 to 63) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 1<br>MBE62 | Off-Line End status (32 to 63) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 2<br>MBE61 | Off-Line End status (32 to 63) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 3<br>MBE60 | Off-Line End status (32 to 63) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 4<br>MBE59 | Off-Line End status (32 to 63) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 5<br>MBE58 | Off-Line End status (32 to 63) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 6<br>MBE57 | Off-Line End status (32 to 63) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 7<br>MBE56 | Off-Line End status (32 to 63) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 8<br>MBE55 | Off-Line End status (32 to 63) of the selected MBIST                                                                  |

Table continues on the next page...



**STCU2\_MBEM field descriptions (continued)**

| <b>Field</b> | <b>Description</b>                                                                                                        |
|--------------|---------------------------------------------------------------------------------------------------------------------------|
|              | 0 MBIST execution still ongoing<br>1 MBIST execution finished                                                             |
| 9<br>MBE54   | Off-Line End status (32 to 63) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 10<br>MBE53  | Off-Line End status (32 to 63) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 11<br>MBE52  | Off-Line End status (32 to 63) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 12<br>MBE51  | Off-Line End status (32 to 63) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 13<br>MBE50  | Off-Line End status (32 to 63) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 14<br>MBE49  | Off-Line End status (32 to 63) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 15<br>MBE48  | Off-Line End status (32 to 63) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 16<br>MBE47  | Off-Line End status (32 to 63) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 17<br>MBE46  | Off-Line End status (32 to 63) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 18<br>MBE45  | Off-Line End status (32 to 63) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 19<br>MBE44  | Off-Line End status (32 to 63) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 20<br>MBE43  | Off-Line End status (32 to 63) of the selected MBIST                                                                      |

*Table continues on the next page...*

**STCU2\_MBEM field descriptions (continued)**

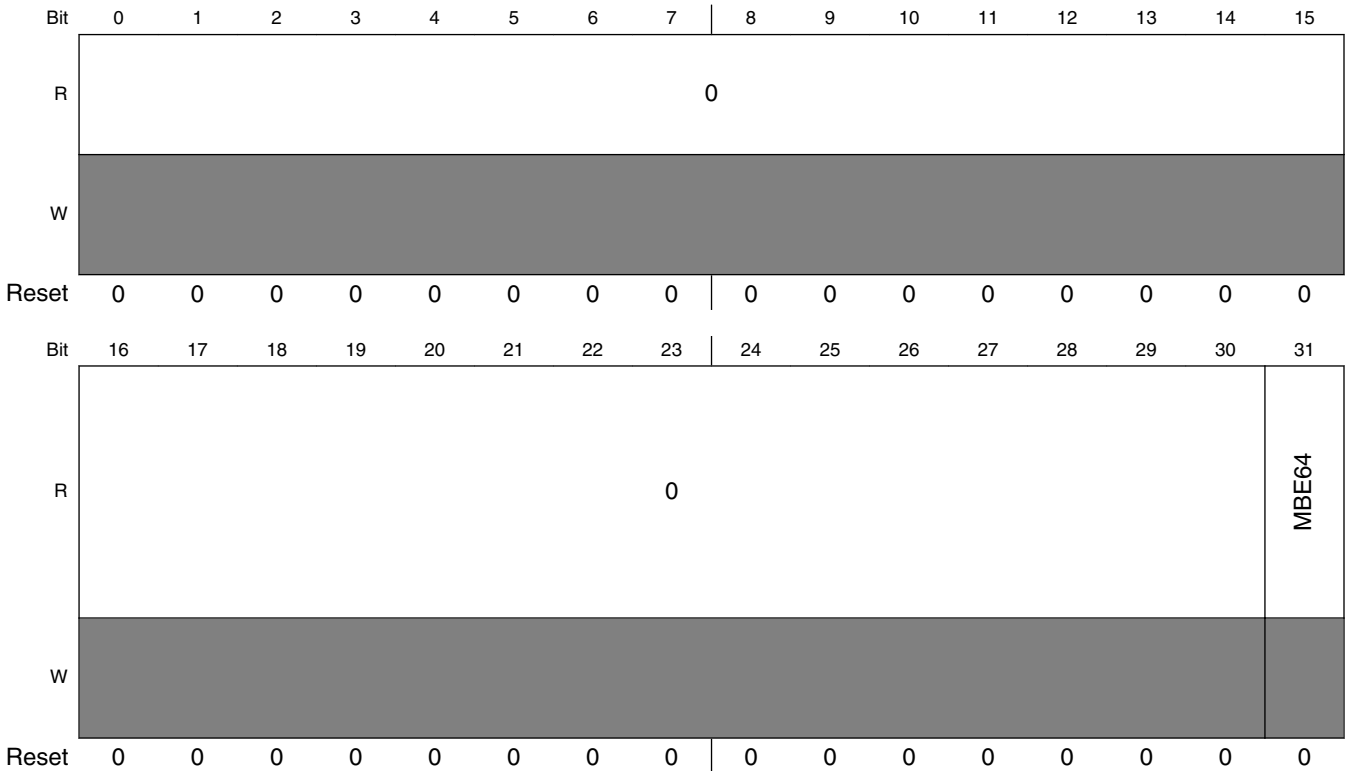
| Field       | Description                                                                                                               |
|-------------|---------------------------------------------------------------------------------------------------------------------------|
|             | 0 MBIST execution still ongoing<br>1 MBIST execution finished                                                             |
| 21<br>MBE42 | Off-Line End status (32 to 63) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 22<br>MBE41 | Off-Line End status (32 to 63) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 23<br>MBE40 | Off-Line End status (32 to 63) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 24<br>MBE39 | Off-Line End status (32 to 63) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 25<br>MBE38 | Off-Line End status (32 to 63) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 26<br>MBE37 | Off-Line End status (32 to 63) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 27<br>MBE36 | Off-Line End status (32 to 63) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 28<br>MBE35 | Off-Line End status (32 to 63) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 29<br>MBE34 | Off-Line End status (32 to 63) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 30<br>MBE33 | Off-Line End status (32 to 63) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 31<br>MBE32 | Off-Line End status (32 to 63) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |

56.9.21 STCU2 Off-Line MBIST End Flag High Register (STCU2\_MBEH)

The STCU2\_MBEH register includes the End Flag related to the execution of the selected Off-Line MBIST in the range NMCUT = 64..95.

The size of the register depends on the number of BISTed RAMs/ROMs.

Address: 0h base + 58h offset = 58h



STCU2\_MBEH field descriptions

| Field            | Description                                                                                                                 |
|------------------|-----------------------------------------------------------------------------------------------------------------------------|
| 0–30<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                     |
| 31<br>MBE64      | MBEx: Off-Line End status (64 to 95) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |

56.9.22 STCU2 On-Line MBIST Status Low Register (STCU2\_MBSLSW)

The STCU2\_MBSLSW register includes the results corresponding to the execution of the selected On-Line MBIST in the range NMCUT = 0-31.

The size of the register depends on the number of BISTed RAMs/ROMs.

The content of this register is initialized to its reset value as soon as STCU2\_RUNSW[RUNSW] = 1.

Address: 0h base + 5Ch offset = 5Ch

|       |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
|-------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| Bit   | 0       | 1       | 2       | 3       | 4       | 5       | 6       | 7       | 8       | 9       | 10      | 11      | 12      | 13      | 14      | 15      |
| R     | MBSSW31 | MBSSW30 | MBSSW29 | MBSSW28 | MBSSW27 | MBSSW26 | MBSSW25 | MBSSW24 | MBSSW23 | MBSSW22 | MBSSW21 | MBSSW20 | MBSSW19 | MBSSW18 | MBSSW17 | MBSSW16 |
| W     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| Reset | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       |

|       |         |         |         |         |         |         |        |        |        |        |        |        |        |        |        |        |
|-------|---------|---------|---------|---------|---------|---------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Bit   | 16      | 17      | 18      | 19      | 20      | 21      | 22     | 23     | 24     | 25     | 26     | 27     | 28     | 29     | 30     | 31     |
| R     | MBSSW15 | MBSSW14 | MBSSW13 | MBSSW12 | MBSSW11 | MBSSW10 | MBSSW9 | MBSSW8 | MBSSW7 | MBSSW6 | MBSSW5 | MBSSW4 | MBSSW3 | MBSSW2 | MBSSW1 | MBSSW0 |
| W     |         |         |         |         |         |         |        |        |        |        |        |        |        |        |        |        |
| Reset | 0       | 0       | 0       | 0       | 0       | 0       | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      |

STCU2\_MBSLSW field descriptions

| Field        | Description                                                  |
|--------------|--------------------------------------------------------------|
| 0<br>MBSSW31 | On-Line status (NMCUT range = 0 to 31) of the selected MBIST |

Table continues on the next page...

**STCU2\_MBSLSW field descriptions (continued)**

| Field         | Description                                                                                                                                              |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
|               | 0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution                                                                     |
| 1<br>MBSSW30  | On-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 2<br>MBSSW29  | On-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 3<br>MBSSW28  | On-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 4<br>MBSSW27  | On-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 5<br>MBSSW26  | On-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 6<br>MBSSW25  | On-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 7<br>MBSSW24  | On-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 8<br>MBSSW23  | On-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 9<br>MBSSW22  | On-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 10<br>MBSSW21 | On-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 11<br>MBSSW20 | On-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 12<br>MBSSW19 | On-Line status (NMCUT range = 0 to 31) of the selected MBIST                                                                                             |

*Table continues on the next page...*

**STCU2\_MBSLSW field descriptions (continued)**

| Field         | Description                                                                                                                                              |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
|               | 0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution                                                                     |
| 13<br>MBSSW18 | On-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 14<br>MBSSW17 | On-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 15<br>MBSSW16 | On-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 16<br>MBSSW15 | On-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 17<br>MBSSW14 | On-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 18<br>MBSSW13 | On-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 19<br>MBSSW12 | On-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 20<br>MBSSW11 | On-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 21<br>MBSSW10 | On-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 22<br>MBSSW9  | On-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 23<br>MBSSW8  | On-Line status (NMCUT range = 0 to 31) of the selected MBIST<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 24<br>MBSSW7  | On-Line status (NMCUT range = 0 to 31) of the selected MBIST                                                                                             |

*Table continues on the next page...*

**STCU2\_MBSLSW field descriptions (continued)**

| Field        | Description                                                                                                                                          |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
|              | 0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution                                                                 |
| 25<br>MBSSW6 | On-Line status (NMCUT range = 0 to 31) of the selected MBIST<br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 26<br>MBSSW5 | On-Line status (NMCUT range = 0 to 31) of the selected MBIST<br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 27<br>MBSSW4 | On-Line status (NMCUT range = 0 to 31) of the selected MBIST<br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 28<br>MBSSW3 | On-Line status (NMCUT range = 0 to 31) of the selected MBIST<br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 29<br>MBSSW2 | On-Line status (NMCUT range = 0 to 31) of the selected MBIST<br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 30<br>MBSSW1 | On-Line status (NMCUT range = 0 to 31) of the selected MBIST<br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 31<br>MBSSW0 | On-Line status (NMCUT range = 0 to 31) of the selected MBIST<br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |

**56.9.23 STCU2 On-Line MBIST Status Medium Register (STCU2\_MBSMSW)**

The STCU2\_MBSMSW register includes the results corresponding to the execution of the selected On-Line MBIST in the range NMCUT = 32..63.

The size of the register depends on the number of BISTed RAMs/ROMs.

The content of this register is initialized to its reset value as soon as STCU2\_RUNSW[RUNSW] = 1.

## Register description

Address: 0h base + 60h offset = 60h

|       |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
|-------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| Bit   | 0       | 1       | 2       | 3       | 4       | 5       | 6       | 7       | 8       | 9       | 10      | 11      | 12      | 13      | 14      | 15      |
| R     | MBSSW63 | MBSSW62 | MBSSW61 | MBSSW60 | MBSSW59 | MBSSW58 | MBSSW57 | MBSSW56 | MBSSW55 | MBSSW54 | MBSSW53 | MBSSW52 | MBSSW51 | MBSSW50 | MBSSW49 | MBSSW48 |
| W     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| Reset | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       |

|       |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
|-------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| Bit   | 16      | 17      | 18      | 19      | 20      | 21      | 22      | 23      | 24      | 25      | 26      | 27      | 28      | 29      | 30      | 31      |
| R     | MBSSW47 | MBSSW46 | MBSSW45 | MBSSW44 | MBSSW43 | MBSSW42 | MBSSW41 | MBSSW40 | MBSSW39 | MBSSW38 | MBSSW37 | MBSSW36 | MBSSW35 | MBSSW34 | MBSSW33 | MBSSW32 |
| W     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| Reset | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       |

### STCU2\_MBSMSW field descriptions

| Field        | Description                                                                                                                                                   |
|--------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>MBSSW63 | MBSSWx: On-Line status (NMCUT range = 32 to 63) of the selected MBIST<br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 1<br>MBSSW62 | MBSSWx: On-Line status (NMCUT range = 32 to 63) of the selected MBIST<br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 2<br>MBSSW61 | MBSSWx: On-Line status (NMCUT range = 32 to 63) of the selected MBIST<br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 3<br>MBSSW60 | MBSSWx: On-Line status (NMCUT range = 32 to 63) of the selected MBIST                                                                                         |

Table continues on the next page...



**STCU2\_MBSMSW field descriptions (continued)**

| <b>Field</b>  | <b>Description</b>                                                                                                                                                |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               | 0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution                                                                              |
| 4<br>MBSSW59  | MBSSWx: On-Line status (NMCUT range = 32 to 63) of the selected MBIST<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 5<br>MBSSW58  | MBSSWx: On-Line status (NMCUT range = 32 to 63) of the selected MBIST<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 6<br>MBSSW57  | MBSSWx: On-Line status (NMCUT range = 32 to 63) of the selected MBIST<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 7<br>MBSSW56  | MBSSWx: On-Line status (NMCUT range = 32 to 63) of the selected MBIST<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 8<br>MBSSW55  | MBSSWx: On-Line status (NMCUT range = 32 to 63) of the selected MBIST<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 9<br>MBSSW54  | MBSSWx: On-Line status (NMCUT range = 32 to 63) of the selected MBIST<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 10<br>MBSSW53 | MBSSWx: On-Line status (NMCUT range = 32 to 63) of the selected MBIST<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 11<br>MBSSW52 | MBSSWx: On-Line status (NMCUT range = 32 to 63) of the selected MBIST<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 12<br>MBSSW51 | MBSSWx: On-Line status (NMCUT range = 32 to 63) of the selected MBIST<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 13<br>MBSSW50 | MBSSWx: On-Line status (NMCUT range = 32 to 63) of the selected MBIST<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 14<br>MBSSW49 | MBSSWx: On-Line status (NMCUT range = 32 to 63) of the selected MBIST<br><br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 15<br>MBSSW48 | MBSSWx: On-Line status (NMCUT range = 32 to 63) of the selected MBIST                                                                                             |

*Table continues on the next page...*

**STCU2\_MBSMSW field descriptions (continued)**

| Field         | Description                                                                                                                                                   |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               | 0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution                                                                          |
| 16<br>MBSSW47 | MBSSWx: On-Line status (NMCUT range = 32 to 63) of the selected MBIST<br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 17<br>MBSSW46 | MBSSWx: On-Line status (NMCUT range = 32 to 63) of the selected MBIST<br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 18<br>MBSSW45 | MBSSWx: On-Line status (NMCUT range = 32 to 63) of the selected MBIST<br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 19<br>MBSSW44 | MBSSWx: On-Line status (NMCUT range = 32 to 63) of the selected MBIST<br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 20<br>MBSSW43 | MBSSWx: On-Line status (NMCUT range = 32 to 63) of the selected MBIST<br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 21<br>MBSSW42 | MBSSWx: On-Line status (NMCUT range = 32 to 63) of the selected MBIST<br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 22<br>MBSSW41 | MBSSWx: On-Line status (NMCUT range = 32 to 63) of the selected MBIST<br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 23<br>MBSSW40 | MBSSWx: On-Line status (NMCUT range = 32 to 63) of the selected MBIST<br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 24<br>MBSSW39 | MBSSWx: On-Line status (NMCUT range = 32 to 63) of the selected MBIST<br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 25<br>MBSSW38 | MBSSWx: On-Line status (NMCUT range = 32 to 63) of the selected MBIST<br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 26<br>MBSSW37 | MBSSWx: On-Line status (NMCUT range = 32 to 63) of the selected MBIST<br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 27<br>MBSSW36 | MBSSWx: On-Line status (NMCUT range = 32 to 63) of the selected MBIST                                                                                         |

*Table continues on the next page...*

**STCU2\_MBSMSW field descriptions (continued)**

| Field         | Description                                                                                                                                                   |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
|               | 0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution                                                                          |
| 28<br>MBSSW35 | MBSSWx: On-Line status (NMCUT range = 32 to 63) of the selected MBIST<br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 29<br>MBSSW34 | MBSSWx: On-Line status (NMCUT range = 32 to 63) of the selected MBIST<br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 30<br>MBSSW33 | MBSSWx: On-Line status (NMCUT range = 32 to 63) of the selected MBIST<br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |
| 31<br>MBSSW32 | MBSSWx: On-Line status (NMCUT range = 32 to 63) of the selected MBIST<br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |

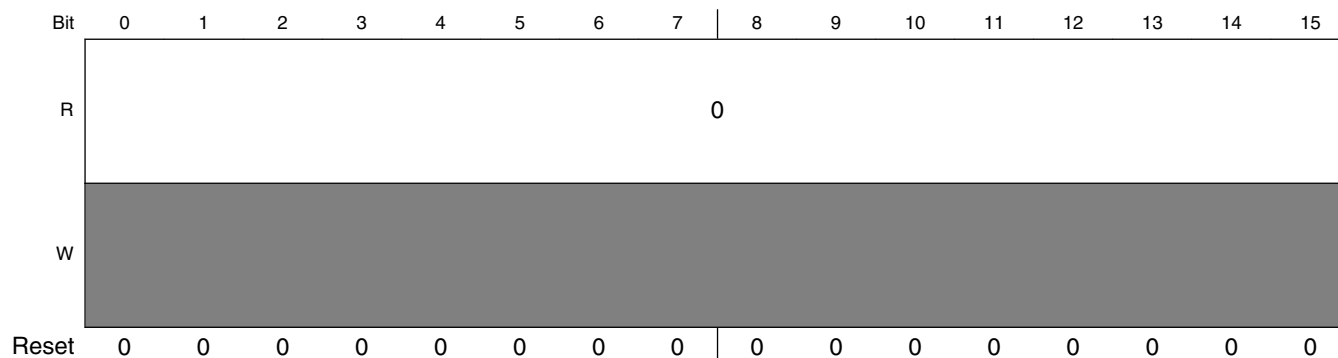
**56.9.24 STCU2 On-Line MBIST Status High Register (STCU2\_MBSHSW)**

The STCU2\_MBSHSW register includes the results corresponding to the execution of the selected On-Line MBIST in the range NMCUT = 64..95.

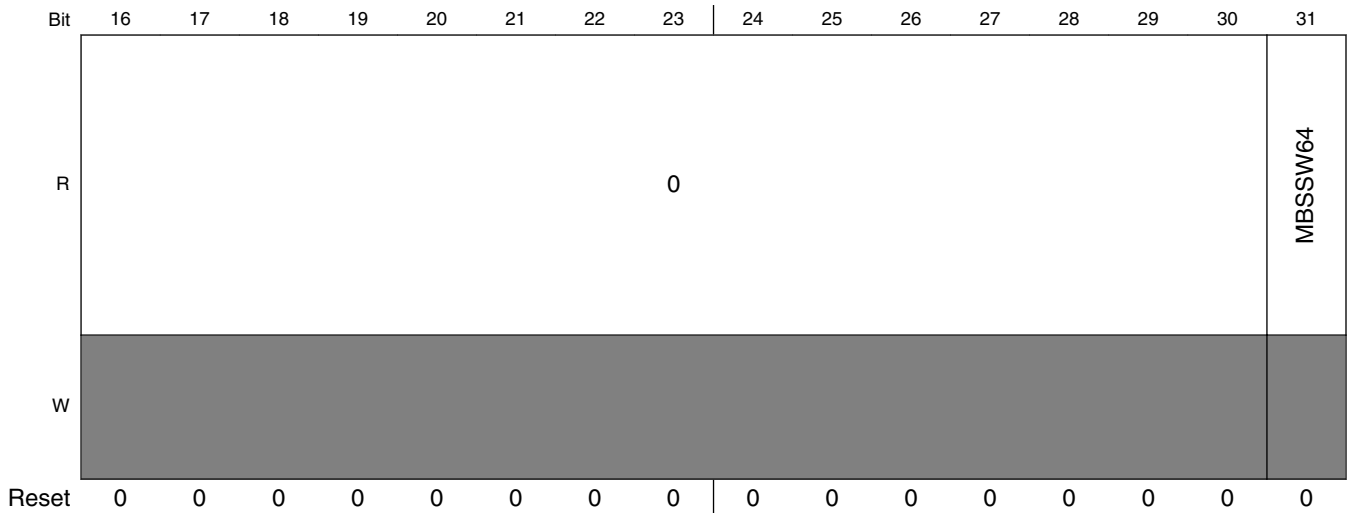
The size of the register depends on the number of BISTed RAMs/ROMs.

The content of this register is initialized to its reset value as soon as STCU2\_RUNSW[RUNSW] = 1.

Address: 0h base + 64h offset = 64h



Register description



STCU2\_MBSSW field descriptions

| Field            | Description                                                                                                                                                   |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–30<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                       |
| 31<br>MBSSW64    | MBSSWx: On-Line status (NMCUT range = 64 to 95) of the selected MBIST<br>0 Failed NMCUT BIST execution<br>1 No Fault detected during the NMCUT BIST execution |

56.9.25 STCU2 On-Line MBIST End Flag Low Register (STCU2\_MBELSW)

The STCU2\_MBELSW register includes the End Flag related to the execution of the selected On-Line MBIST in the range NMCUT = 0-31.

The size of the register depends on the number of BISTed RAMs/ROMs.

The content of this register is initialized to its reset value as soon as STCU2\_RUNSW[RUNSW] = 1.

Address: 0h base + 68h offset = 68h

|       |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
|-------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| Bit   | 0       | 1       | 2       | 3       | 4       | 5       | 6       | 7       | 8       | 9       | 10      | 11      | 12      | 13      | 14      | 15      |
| R     | MBESW31 | MBESW30 | MBESW29 | MBESW28 | MBESW27 | MBESW26 | MBESW25 | MBESW24 | MBESW23 | MBESW22 | MBESW21 | MBESW20 | MBESW19 | MBESW18 | MBESW17 | MBESW16 |
| W     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| Reset | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       |

|       |         |         |         |         |         |         |        |        |        |        |        |        |        |        |        |        |
|-------|---------|---------|---------|---------|---------|---------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Bit   | 16      | 17      | 18      | 19      | 20      | 21      | 22     | 23     | 24     | 25     | 26     | 27     | 28     | 29     | 30     | 31     |
| R     | MBESW15 | MBESW14 | MBESW13 | MBESW12 | MBESW11 | MBESW10 | MBESW9 | MBESW8 | MBESW7 | MBESW6 | MBESW5 | MBESW4 | MBESW3 | MBESW2 | MBESW1 | MBESW0 |
| W     |         |         |         |         |         |         |        |        |        |        |        |        |        |        |        |        |
| Reset | 0       | 0       | 0       | 0       | 0       | 0       | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 0      |

**STCU2\_MBELSW field descriptions**

| Field        | Description                                                                                                                 |
|--------------|-----------------------------------------------------------------------------------------------------------------------------|
| 0<br>MBESW31 | MBESWx: On-Line End status (0 to 31) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 1<br>MBESW30 | MBESWx: On-Line End status (0 to 31) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 2<br>MBESW29 | MBESWx: On-Line End status (0 to 31) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 3<br>MBESW28 | MBESWx: On-Line End status (0 to 31) of the selected MBIST                                                                  |

*Table continues on the next page...*

**STCU2\_MBESW field descriptions (continued)**

| Field         | Description                                                                                                                     |
|---------------|---------------------------------------------------------------------------------------------------------------------------------|
|               | 0 MBIST execution still ongoing<br>1 MBIST execution finished                                                                   |
| 4<br>MBESW27  | MBESWx: On-Line End status (0 to 31) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 5<br>MBESW26  | MBESWx: On-Line End status (0 to 31) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 6<br>MBESW25  | MBESWx: On-Line End status (0 to 31) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 7<br>MBESW24  | MBESWx: On-Line End status (0 to 31) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 8<br>MBESW23  | MBESWx: On-Line End status (0 to 31) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 9<br>MBESW22  | MBESWx: On-Line End status (0 to 31) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 10<br>MBESW21 | MBESWx: On-Line End status (0 to 31) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 11<br>MBESW20 | MBESWx: On-Line End status (0 to 31) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 12<br>MBESW19 | MBESWx: On-Line End status (0 to 31) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 13<br>MBESW18 | MBESWx: On-Line End status (0 to 31) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 14<br>MBESW17 | MBESWx: On-Line End status (0 to 31) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 15<br>MBESW16 | MBESWx: On-Line End status (0 to 31) of the selected MBIST                                                                      |

*Table continues on the next page...*

**STCU2\_MBESW field descriptions (continued)**

| Field         | Description                                                                                                                     |
|---------------|---------------------------------------------------------------------------------------------------------------------------------|
|               | 0 MBIST execution still ongoing<br>1 MBIST execution finished                                                                   |
| 16<br>MBESW15 | MBESWx: On-Line End status (0 to 31) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 17<br>MBESW14 | MBESWx: On-Line End status (0 to 31) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 18<br>MBESW13 | MBESWx: On-Line End status (0 to 31) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 19<br>MBESW12 | MBESWx: On-Line End status (0 to 31) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 20<br>MBESW11 | MBESWx: On-Line End status (0 to 31) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 21<br>MBESW10 | MBESWx: On-Line End status (0 to 31) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 22<br>MBESW9  | MBESWx: On-Line End status (0 to 31) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 23<br>MBESW8  | MBESWx: On-Line End status (0 to 31) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 24<br>MBESW7  | MBESWx: On-Line End status (0 to 31) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 25<br>MBESW6  | MBESWx: On-Line End status (0 to 31) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 26<br>MBESW5  | MBESWx: On-Line End status (0 to 31) of the selected MBIST<br><br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 27<br>MBESW4  | MBESWx: On-Line End status (0 to 31) of the selected MBIST                                                                      |

*Table continues on the next page...*

**STCU2\_MBELSW field descriptions (continued)**

| Field        | Description                                                                                                                 |
|--------------|-----------------------------------------------------------------------------------------------------------------------------|
|              | 0 MBIST execution still ongoing<br>1 MBIST execution finished                                                               |
| 28<br>MBESW3 | MBESWx: On-Line End status (0 to 31) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 29<br>MBESW2 | MBESWx: On-Line End status (0 to 31) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 30<br>MBESW1 | MBESWx: On-Line End status (0 to 31) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 31<br>MBESW0 | MBESWx: On-Line End status (0 to 31) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |

**56.9.26 STCU2 On-Line MBIST End Flag Medium Register (STCU2\_MBEMSW)**

The STCU2\_MBEMSW register includes the End Flag related to the execution of the selected On-Line MBIST in the range NMCUT = 32-63.

The size of the register depends on the number of BISTed RAMs/ROMs.

The content of this register is initialized to its reset value as soon as STCU2\_RUNSW[RUNSW] = 1.

Address: 0h base + 6Ch offset = 6Ch

| Bit   | 0       | 1       | 2       | 3       | 4       | 5       | 6       | 7       | 8       | 9       | 10      | 11      | 12      | 13      | 14      | 15      |
|-------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| R     | MBESW63 | MBESW62 | MBESW61 | MBESW60 | MBESW59 | MBESW58 | MBESW57 | MBESW56 | MBESW55 | MBESW54 | MBESW53 | MBESW52 | MBESW51 | MBESW50 | MBESW49 | MBESW48 |
| W     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| Reset | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       |



|       |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
|-------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| Bit   | 16      | 17      | 18      | 19      | 20      | 21      | 22      | 23      | 24      | 25      | 26      | 27      | 28      | 29      | 30      | 31      |
| R     | MBESW47 | MBESW46 | MBESW45 | MBESW44 | MBESW43 | MBESW42 | MBESW41 | MBESW40 | MBESW39 | MBESW38 | MBESW37 | MBESW36 | MBESW35 | MBESW34 | MBESW33 | MBESW32 |
| W     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| Reset | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       | 0       |

**STCU2\_MBEMSW field descriptions**

| Field        | Description                                                                                                          |
|--------------|----------------------------------------------------------------------------------------------------------------------|
| 0<br>MBESW63 | On-Line End status (32 to 63) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 1<br>MBESW62 | On-Line End status (32 to 63) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 2<br>MBESW61 | On-Line End status (32 to 63) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 3<br>MBESW60 | On-Line End status (32 to 63) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 4<br>MBESW59 | On-Line End status (32 to 63) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 5<br>MBESW58 | On-Line End status (32 to 63) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 6<br>MBESW57 | On-Line End status (32 to 63) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 7<br>MBESW56 | On-Line End status (32 to 63) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |

*Table continues on the next page...*

**STCU2\_MBEMSW field descriptions (continued)**

| Field         | Description                                                                                                          |
|---------------|----------------------------------------------------------------------------------------------------------------------|
| 8<br>MBESW55  | On-Line End status (32 to 63) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 9<br>MBESW54  | On-Line End status (32 to 63) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 10<br>MBESW53 | On-Line End status (32 to 63) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 11<br>MBESW52 | On-Line End status (32 to 63) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 12<br>MBESW51 | On-Line End status (32 to 63) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 13<br>MBESW50 | On-Line End status (32 to 63) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 14<br>MBESW49 | On-Line End status (32 to 63) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 15<br>MBESW48 | On-Line End status (32 to 63) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 16<br>MBESW47 | On-Line End status (32 to 63) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 17<br>MBESW46 | On-Line End status (32 to 63) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 18<br>MBESW45 | On-Line End status (32 to 63) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 19<br>MBESW44 | On-Line End status (32 to 63) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |

*Table continues on the next page...*

**STCU2\_MBEMSW field descriptions (continued)**

| <b>Field</b>  | <b>Description</b>                                                                                                   |
|---------------|----------------------------------------------------------------------------------------------------------------------|
| 20<br>MBESW43 | On-Line End status (32 to 63) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 21<br>MBESW42 | On-Line End status (32 to 63) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 22<br>MBESW41 | On-Line End status (32 to 63) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 23<br>MBESW40 | On-Line End status (32 to 63) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 24<br>MBESW39 | On-Line End status (32 to 63) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 25<br>MBESW38 | On-Line End status (32 to 63) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 26<br>MBESW37 | On-Line End status (32 to 63) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 27<br>MBESW36 | On-Line End status (32 to 63) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 28<br>MBESW35 | On-Line End status (32 to 63) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 29<br>MBESW34 | On-Line End status (32 to 63) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 30<br>MBESW33 | On-Line End status (32 to 63) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |
| 31<br>MBESW32 | On-Line End status (32 to 63) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |

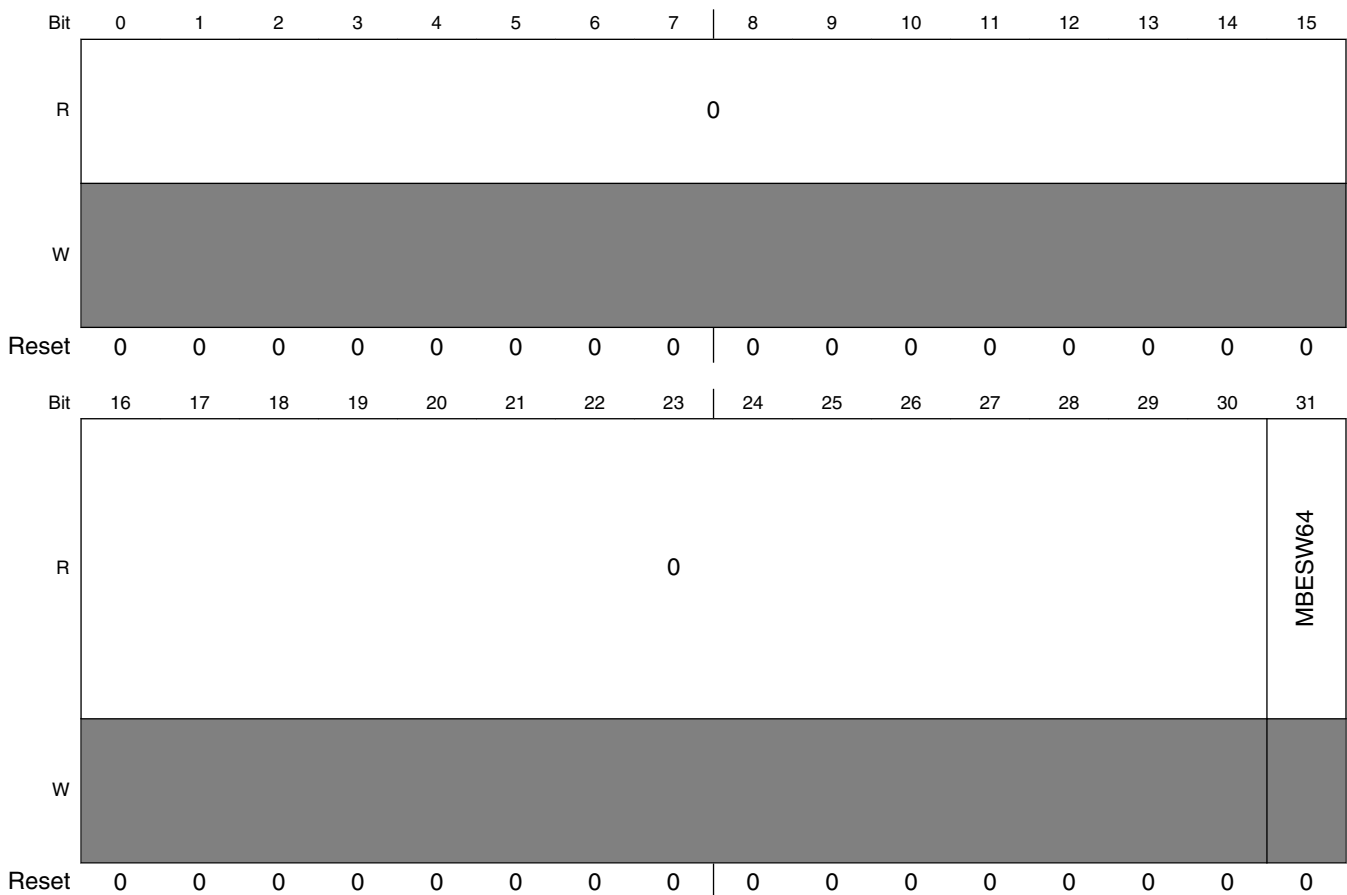
56.9.27 STCU2 On-Line MBIST End Flag High Register (STCU2\_MBEHSW)

The STCU2\_MBEHSW register includes the End Flag related to the execution of the selected On-Line MBIST in the range NMCUT = 64-95.

The size of the register depends on the number of BISTed RAMs/ROMs.

The content of this register is initialized to its reset value as soon as STCU2\_RUNSW[RUNSW] = 1.

Address: 0h base + 70h offset = 70h



STCU2\_MBEHSW field descriptions

| Field            | Description                                                                                                                  |
|------------------|------------------------------------------------------------------------------------------------------------------------------|
| 0–30<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                      |
| 31<br>MBESW64    | MBESWx: On-Line End status (64 to 95) of the selected MBIST<br>0 MBIST execution still ongoing<br>1 MBIST execution finished |

## 56.9.28 STCU2 MBIST Unrecoverable FM Low Register (STCU2\_MBUFML)

### NOTE

The reset value is chip-specific; see the chapter that describes how modules are configured and connected.

The STCU2\_MBUFML register defines the fault mapping, in terms of Unrecoverable or Recoverable fault, of the MBIST in the range NMCUT = 0-31.

The size of the register depends on the number of BISTed RAMs/ROMs.

The R/W fields in this register are readable at any time. You can write to these fields when either of the following conditions is true:

- Off-line Self-Test phase is active
- On-line Self-Test phase is active and STCU2\_CFG[WRP] = 0

Address: 0h base + 74h offset = 74h

| Bit   | 0       | 1       | 2       | 3       | 4       | 5       | 6       | 7       | 8       | 9       | 10      | 11      | 12      | 13      | 14      | 15      |
|-------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| R     | MBUFM31 | MBUFM30 | MBUFM29 | MBUFM28 | MBUFM27 | MBUFM26 | MBUFM25 | MBUFM24 | MBUFM23 | MBUFM22 | MBUFM21 | MBUFM20 | MBUFM19 | MBUFM18 | MBUFM17 | MBUFM16 |
| W     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| Reset | x*      | x*      | x*      | x*      | x*      | x*      | x*      | x*      | x*      | x*      | x*      | x*      | x*      | x*      | x*      | x*      |

| Bit   | 16      | 17      | 18      | 19      | 20      | 21      | 22     | 23     | 24     | 25     | 26     | 27     | 28     | 29     | 30     | 31     |
|-------|---------|---------|---------|---------|---------|---------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| R     | MBUFM15 | MBUFM14 | MBUFM13 | MBUFM12 | MBUFM11 | MBUFM10 | MBUFM9 | MBUFM8 | MBUFM7 | MBUFM6 | MBUFM5 | MBUFM4 | MBUFM3 | MBUFM2 | MBUFM1 | MBUFM0 |
| W     |         |         |         |         |         |         |        |        |        |        |        |        |        |        |        |        |
| Reset | x*      | x*      | x*      | x*      | x*      | x*      | x*     | x*     | x*     | x*     | x*     | x*     | x*     | x*     | x*     | x*     |

\* Notes:

- x = Undefined at reset.

### STCU2\_MBUFML field descriptions

| Field        | Description                                                                                       |
|--------------|---------------------------------------------------------------------------------------------------|
| 0<br>MBUFM31 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 1<br>MBUFM30 | MBIST Unrecoverable Fault Mapping                                                                 |

Table continues on the next page...

**STCU2\_MBUFML field descriptions (continued)**

| Field         | Description                                                                                       |
|---------------|---------------------------------------------------------------------------------------------------|
|               | 0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping                                      |
| 2<br>MBUFM29  | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 3<br>MBUFM28  | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 4<br>MBUFM27  | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 5<br>MBUFM26  | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 6<br>MBUFM25  | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 7<br>MBUFM24  | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 8<br>MBUFM23  | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 9<br>MBUFM22  | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 10<br>MBUFM21 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 11<br>MBUFM20 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 12<br>MBUFM19 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 13<br>MBUFM18 | MBIST Unrecoverable Fault Mapping                                                                 |

*Table continues on the next page...*

**STCU2\_MBUFML field descriptions (continued)**

| Field         | Description                                                                                       |
|---------------|---------------------------------------------------------------------------------------------------|
|               | 0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping                                      |
| 14<br>MBUFM17 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 15<br>MBUFM16 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 16<br>MBUFM15 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 17<br>MBUFM14 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 18<br>MBUFM13 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 19<br>MBUFM12 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 20<br>MBUFM11 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 21<br>MBUFM10 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 22<br>MBUFM9  | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 23<br>MBUFM8  | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 24<br>MBUFM7  | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 25<br>MBUFM6  | MBIST Unrecoverable Fault Mapping                                                                 |

*Table continues on the next page...*

**STCU2\_MBUFML field descriptions (continued)**

| Field        | Description                                                                                       |
|--------------|---------------------------------------------------------------------------------------------------|
|              | 0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping                                      |
| 26<br>MBUFM5 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 27<br>MBUFM4 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 28<br>MBUFM3 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 29<br>MBUFM2 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 30<br>MBUFM1 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 31<br>MBUFM0 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |

**56.9.29 STCU2 MBIST Unrecoverable FM Medium Register (STCU2\_MBUFMM)****NOTE**

The reset value is chip-specific; see the chapter that describes how modules are configured and connected.

The STCU2\_MBUFMM register defines the fault mapping, in terms of Unrecoverable or Recoverable fault, of the MBIST in the range NMCUT = 32-63.

The size of the register depends on the number of BISTed RAMs/ROMs.

The R/W fields in this register are readable at any time. You can write to these fields when either of the following conditions is true:

- Off-line Self-Test phase is active
- On-line Self-Test phase is active and STCU2\_CFG[WRP] = 0



Address: 0h base + 78h offset = 78h

|       |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
|-------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| Bit   | 0       | 1       | 2       | 3       | 4       | 5       | 6       | 7       | 8       | 9       | 10      | 11      | 12      | 13      | 14      | 15      |
| R     | MBUFM63 | MBUFM62 | MBUFM61 | MBUFM60 | MBUFM59 | MBUFM58 | MBUFM57 | MBUFM56 | MBUFM55 | MBUFM54 | MBUFM53 | MBUFM52 | MBUFM51 | MBUFM50 | MBUFM49 | MBUFM48 |
| W     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| Reset | x*      | x*      | x*      | x*      | x*      | x*      | x*      | x*      | x*      | x*      | x*      | x*      | x*      | x*      | x*      | x*      |

|       |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
|-------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| Bit   | 16      | 17      | 18      | 19      | 20      | 21      | 22      | 23      | 24      | 25      | 26      | 27      | 28      | 29      | 30      | 31      |
| R     | MBUFM47 | MBUFM46 | MBUFM45 | MBUFM44 | MBUFM43 | MBUFM42 | MBUFM41 | MBUFM40 | MBUFM39 | MBUFM38 | MBUFM37 | MBUFM36 | MBUFM35 | MBUFM34 | MBUFM33 | MBUFM32 |
| W     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |
| Reset | x*      | x*      | x*      | x*      | x*      | x*      | x*      | x*      | x*      | x*      | x*      | x*      | x*      | x*      | x*      | x*      |

\* Notes:

- x = Undefined at reset.

## STCU2\_MBUFMM field descriptions

| Field        | Description                                                                                       |
|--------------|---------------------------------------------------------------------------------------------------|
| 0<br>MBUFM63 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 1<br>MBUFM62 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 2<br>MBUFM61 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 3<br>MBUFM60 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 4<br>MBUFM59 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 5<br>MBUFM58 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 6<br>MBUFM57 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |

Table continues on the next page...

**STCU2\_MBUFMM field descriptions (continued)**

| Field         | Description                                                                                       |
|---------------|---------------------------------------------------------------------------------------------------|
| 7<br>MBUFM56  | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 8<br>MBUFM55  | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 9<br>MBUFM54  | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 10<br>MBUFM53 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 11<br>MBUFM52 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 12<br>MBUFM51 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 13<br>MBUFM50 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 14<br>MBUFM49 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 15<br>MBUFM48 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 16<br>MBUFM47 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 17<br>MBUFM46 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 18<br>MBUFM45 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |

*Table continues on the next page...*

**STCU2\_MBUFMM field descriptions (continued)**

| <b>Field</b>  | <b>Description</b>                                                                                |
|---------------|---------------------------------------------------------------------------------------------------|
| 19<br>MBUFM44 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 20<br>MBUFM43 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 21<br>MBUFM42 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 22<br>MBUFM41 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 23<br>MBUFM40 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 24<br>MBUFM39 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 25<br>MBUFM38 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 26<br>MBUFM37 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 27<br>MBUFM36 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 28<br>MBUFM35 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 29<br>MBUFM34 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |
| 30<br>MBUFM33 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |

*Table continues on the next page...*

**STCU2\_MBUFMM field descriptions (continued)**

| Field         | Description                                                                                       |
|---------------|---------------------------------------------------------------------------------------------------|
| 31<br>MBUFM32 | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |

**56.9.30 STCU2 MBIST Unrecoverable FM High Register (STCU2\_MBUFMMH)****NOTE**

The reset value is chip-specific; see the chapter that describes how modules are configured and connected.

The STCU2\_MBUFMMH register defines the fault mapping, in terms of Unrecoverable or Recoverable fault, of the MBIST in the range NMCUT = 64-95.

The size of the register depends on the number of BISTed RAMs/ROMs.

The R/W fields in this register are readable at any time. You can write to these fields when either of the following conditions is true:

- Off-line Self-Test phase is active
- On-line Self-Test phase is active and STCU2\_CFG[WRP] = 0

Address: 0h base + 7Ch offset = 7Ch

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---------|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15      |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |
| Reset | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x*      |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31      |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    | MBUFM64 |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |         |
| Reset | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x*      |

\* Notes:

- x = Undefined at reset.

**STCU2\_MBUFMH field descriptions**

| Field            | Description                                                                                       |
|------------------|---------------------------------------------------------------------------------------------------|
| 0–30<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.           |
| 31<br>MBUFM64    | MBIST Unrecoverable Fault Mapping<br>0 Recoverable Fault mapping<br>1 Unrecoverable Fault mapping |

**56.9.31 STCU2 LBIST Control (STCU2\_LB\_CTRLn)****NOTE**

The reset value is chip-specific; see the chapter that describes how modules are configured and connected.

The STCU2\_LB\_CTRL register defines the control setting of each LBIST controller.

The R/W fields in this register are readable at any time. You can write to these fields when either of the following conditions is true:

- Offline self test phase is active
- Online self test phase is active and STCU2\_CFG[WRP] = 0

Address: 0h base + 100h offset + (64d × i), where i=0d to 5d

|       |     |    |    |    |    |    |    |    |    |    |          |          |     |    |    |    |
|-------|-----|----|----|----|----|----|----|----|----|----|----------|----------|-----|----|----|----|
| Bit   | 0   | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10       | 11       | 12  | 13 | 14 | 15 |
| R     | CSM |    |    |    |    |    |    | 0  |    |    | Reserved | Reserved | SHS |    |    |    |
| W     |     |    |    |    |    |    |    |    |    |    |          |          |     |    |    |    |
| Reset | x*  | x* | x* | x* | x* | x* | x* | x* | x* | x* | x*       | x*       | x*  | x* | x* | x* |

|       |          |    |    |    |         |    |    |    |    |    |    |    |     |     |    |    |
|-------|----------|----|----|----|---------|----|----|----|----|----|----|----|-----|-----|----|----|
| Bit   | 16       | 17 | 18 | 19 | 20      | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28  | 29  | 30 | 31 |
| R     | SCEN_OFF |    |    |    | SCEN_ON |    |    |    | 0  |    |    |    | PFT | CWS |    |    |
| W     |          |    |    |    |         |    |    |    |    |    |    |    |     |     |    |    |
| Reset | x*       | x* | x* | x* | x*      | x* | x* | x* | x* | x* | x* | x* | x*  | x*  | x* | x* |

\* Notes:

- x = Undefined at reset.

**STCU2\_LB\_CTRLn field descriptions**

| Field    | Description                                                                                                                                                                                                     |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>CSM | Concurrent/sequential mode<br>The next LBIST is scheduled concurrently to the current one if the CSM bit is set to 1; otherwise, it is scheduled sequentially to the completion of the current LBIST execution. |

*Table continues on the next page...*

## STCU2\_LB\_CTRLn field descriptions (continued)

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | 0 Sequential mode<br>1 Concurrent mode                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 1–7<br>PTR        | <p>Next LBIST pointer</p> <p>PTR defines the logical pointer to the next LBIST. In case of NIL pointer the CSM bit has to be set Sequential (0) to define this is the end of the list. The self testing procedure is stopped when the current LBIST has been completed.</p> <p>LBIST POINTERS</p> <p>0000b - pointer to LBIST partition 0</p> <p>0001b - pointer to LBIST partition 1</p> <p>0010b - pointer to LBIST partition 2</p> <p>0011b - pointer to LBIST partition 3</p> <p>0100b - pointer to LBIST partition 4</p> <p>0101b - pointer to LBIST partition 5</p> <p>...</p> <p>0x7F - pointer to NIL, end of LBIST execution</p> <p>Values not defined in this list will result in an error being set in the STCU2_ERR register.</p> |
| 8–10<br>Reserved  | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 11<br>Reserved    | <p>This field is reserved.</p> <p><b>NOTE:</b> This reserved field is writeable but do not write any value to it other than its reset value</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 12<br>Reserved    | <p>This field is reserved.</p> <p><b>NOTE:</b> This reserved field is writeable but do not write any value to it other than its reset value</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 13–15<br>SHS      | <p>Shift speed</p> <p>SHS defines the shift speed</p> <p>000 Shift at full rate of STCU2 core clock</p> <p>001 Shift at 1/2 rate of STCU2 core clock</p> <p>010 Shift at 1/3 rate of STCU2 core clock</p> <p>011 Shift at 1/4 rate of STCU2 core clock</p> <p>100 Shift at 1/5 rate of STCU2 core clock</p> <p>101 Shift at 1/6 rate of STCU2 core clock</p> <p>110 Shift at 1/7 rate of STCU2 core clock</p> <p>111 Shift at 1/8 rate of STCU2 core clock</p>                                                                                                                                                                                                                                                                                |
| 16–19<br>SCEN_OFF | <p>Scan enable OFF</p> <p>SCEN_OFF defines the number of clock cycles OFF following the falling transition on the SCEN</p> <p><b>NOTE:</b> Scen_off must be programmed to a value &gt;=1</p> <p>0000 0 delay cycles</p> <p>0001 1 delay cycle</p> <p>0010 2 delay cycles</p> <p>0011 3 delay cycles</p> <p>0100 4 delay cycles</p>                                                                                                                                                                                                                                                                                                                                                                                                            |

Table continues on the next page...

## STCU2\_LB\_CTRLn field descriptions (continued)

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                   | 0101 5 delay cycles<br>0110 6 delay cycles<br>0111 7 delay cycles<br>1000 8 delay cycles<br>1001 9 delay cycles<br>1010 10 delay cycles<br>1011 11 delay cycles<br>1100 12 delay cycles<br>1101 13 delay cycles<br>1110 14 delay cycles<br>1111 15 delay cycles                                                                                                                                                                                                                                                                                                                       |
| 20–23<br>SCEN_ON  | Scan enable ON<br>SCEN_ON defines the number of clock cycles OFF following the rising transition on the SCEN<br><b>NOTE:</b> Scen_on delay register value must be programmed to a value $\geq 1$<br>0000 0 delay cycles<br>0001 1 delay cycle<br>0010 2 delay cycles<br>0011 3 delay cycles<br>0100 4 delay cycles<br>0101 5 delay cycles<br>0110 6 delay cycles<br>0111 7 delay cycles<br>1000 8 delay cycles<br>1001 9 delay cycles<br>1010 10 delay cycles<br>1011 11 delay cycles<br>1100 12 delay cycles<br>1101 13 delay cycles<br>1110 14 delay cycles<br>1111 15 delay cycles |
| 24–27<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 28<br>PFT         | Past Flush Test<br>The LBIST controller, by default, applies 32 Flush Test patterns. This bit allows to skip/apply the Flush test pattern sequence.<br>0 Apply Flush Test Patterns<br>1 Skip Flush Test Patterns                                                                                                                                                                                                                                                                                                                                                                      |
| 29–31<br>CWS      | Capture window size<br>CWS defines the capture window size.<br>000 illegal<br>001 controller waits 1 shift cycle for capture to finish<br>010 controller waits 2 shift cycles for capture to finish<br>011 controller waits 3 shift cycles for capture to finish                                                                                                                                                                                                                                                                                                                      |

Table continues on the next page...

**STCU2\_LB\_CTRLn field descriptions (continued)**

| Field | Description                                           |
|-------|-------------------------------------------------------|
| 100   | controller waits 4 shift cycles for capture to finish |
| 101   | controller waits 5 shift cycles for capture to finish |
| 110   | controller waits 6 shift cycles for capture to finish |
| 111   | controller waits 7 shift cycles for capture to finish |

**56.9.32 STCU2 LBIST PC Stop Register (STCU2\_LB\_PCSn)****NOTE**

The reset value is chip-specific; see the chapter that describes how modules are configured and connected.

The STCU2\_LB\_PCS register defines the pattern counter stop of each LBIST controller.

The R/W fields in this register are readable at any time. You can write to these fields when either of the following conditions is true:

- Off-line Self-Test phase is active
- On-line Self-Test phase is active and STCU2\_CFG[WRP] = 0

Address: 0h base + 104h offset + (64d × i), where i=0d to 5d

|       |    |    |    |    |    |    |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----|----|----|----|----|----|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6   | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0  |    |    |    |    |    | PCS |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |    |    |    |    |    |    |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x* | x* | x* | x* | x* | x* | x*  | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

**STCU2\_LB\_PCSn field descriptions**

| Field           | Description                                                                             |
|-----------------|-----------------------------------------------------------------------------------------|
| 0–5<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 6–31<br>PCS     | Pattern counter stop<br>PCS defines the pattern counter stop value.                     |

**56.9.33 STCU2 Off-Line LBIST MISR Expected Low Register (STCU2\_LB\_MISRELn)****NOTE**

The reset value is chip-specific; see the chapter that describes how modules are configured and connected.



The STCU2\_LB\_MISREL register defines the LSB part (31-0) of the Expected MISR of the Off-Line LBIST controller.

The size of the register depends on the number of MISR bits of the related LBIST.

The R/W fields in this register are readable at any time. However, you can write to these fields only when Off-line Self-Test phase is still active.

Address: 0h base + 110h offset + (64d × i), where i=0d to 5d

|       |        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|--------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0      | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | MISREx |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*     | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |

\* Notes:

- x = Undefined at reset.

### STCU2\_LB\_MISREL<sub>n</sub> field descriptions

| Field          | Description                                                                                      |
|----------------|--------------------------------------------------------------------------------------------------|
| 0–31<br>MISREx | Off-Line MISR Expected low Bits<br>This field defines the low part (31..0) of the Expected MISR. |

## 56.9.34 STCU2 Off-Line LBIST MISR Expected High Register (STCU2\_LB\_MISREH<sub>n</sub>)

### NOTE

The reset value is chip-specific; see the chapter that describes how modules are configured and connected.

The STCU2\_LB\_MISREH register defines the MSB part (63-32) of the Expected MISR of the Off-Line LBIST controller.

The size of the register depends on the number of MISR bits of the related LBIST.

The R/W fields in this register are readable at any time. However, you can write to these fields only when Off-line Self-Test phase is still active.

Address: 0h base + 114h offset + (64d × i), where i=0d to 5d

|       |        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|--------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0      | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | MISREx |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*     | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |

\* Notes:

- x = Undefined at reset.

**STCU2\_LB\_MISREH<sub>n</sub> field descriptions**

| Field          | Description                                                                                        |
|----------------|----------------------------------------------------------------------------------------------------|
| 0–31<br>MISREx | Off-Line MISR Expected high Bits<br>This field defines the high part (63-32) of the Expected MISR. |

**56.9.35 STCU2 Off-Line LBIST MISR Read Low Register (STCU2\_LB\_MISRRL<sub>n</sub>)****NOTE**

The reset value is chip-specific; see the chapter that describes how modules are configured and connected.

The STCU2\_LB\_MISRRL register reports the LSB part (31-0) of the MISR obtained at the end of the Off-Line LBIST controller execution.

The size of the register depends on the number of MISR bits of the related LBIST.

Address: 0h base + 118h offset + (64d × i), where i=0d to 5d

|       |         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0       | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | MISRRLx |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | X*      | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* | X* |

\* Notes:

- x = Undefined at reset.

**STCU2\_LB\_MISRRL<sub>n</sub> field descriptions**

| Field           | Description                                                                                                                                    |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>MISRRLx | Off-Line MISR Read Low Bits<br>This field is equivalent to the Low Bits (31-0) of the MISR obtained at the end of the Off-Line LBIST execution |

**56.9.36 STCU2 Off-Line LBIST MISR Read High Register (STCU2\_LB\_MISRRLH<sub>n</sub>)****NOTE**

The reset value is chip-specific; see the chapter that describes how modules are configured and connected.

The STCU2\_LB\_MISRRLH register report the MSB part (63-32) of the MISR obtained at the end of each Off-Line LBIST controller execution.

The size of the register depends on the number of MISR bits of the related LBIST.

Address: 0h base + 11Ch offset + (64d × i), where i=0d to 5d

|       |        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|--------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0      | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | MISRRx |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*     | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

### STCU2\_LB\_MISRRHn field descriptions

| Field          | Description                                                                                                                                           |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>MISRRx | Off-Line MISR Read High Bits<br><br>This field is equivalent to the High Bits (63-32) of the MISR obtained at the end of the Off-Line LBIST execution |

## 56.9.37 STCU2 On-Line LBIST MISR Expected Low Register (STCU2\_LB\_MISRELSWn)

### NOTE

The reset value is chip-specific; see the chapter that describes how modules are configured and connected.

The STCU2\_LB\_MISRELSW register defines the LSB part (31-0) of the Expected MISR of the On-Line LBIST controller.

The size of the register depends on the number of MISR bits of the related LBIST.

The R/W fields in this register are readable at any time. You can write to these fields when either of the following conditions is true:

- Off-line Self-Test phase is active
- On-line Self-Test phase is active and STCU2\_CFG[WRP] = 0

Address: 0h base + 120h offset + (64d × i), where i=0d to 5d

|       |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0        | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | MISRESWx |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*       | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

**STCU2\_LB\_MISRELSW<sub>n</sub> field descriptions**

| Field                        | Description                                                                                     |
|------------------------------|-------------------------------------------------------------------------------------------------|
| 0–31<br>MISRESW <sub>x</sub> | On-Line MISR Expected low Bits<br>This field defines the low part (31..0) of the Expected MISR. |

**56.9.38 STCU2 On-Line LBIST MISR Expected High Register (STCU2\_LB\_MISREHSW<sub>n</sub>)****NOTE**

The reset value is chip-specific; see the chapter that describes how modules are configured and connected.

The STCU2\_LB\_MISREHSW register defines the MSB part (63-32) of the Expected MISR of the On-Line LBIST controller.

The size of the register depends on the number of MISR bits of the related LBIST.

The R/W fields in this register are readable at any time. You can write to these fields when either of the following conditions is true:

- Off-line Self-Test phase is active
- On-line Self-Test phase is active and STCU2\_CFG[WRP] = 0

Address: 0h base + 124h offset + (64d × i), where i=0d to 5d

|       |                      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0                    | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | MISRESW <sub>x</sub> |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |                      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*                   | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |

\* Notes:

- x = Undefined at reset.

**STCU2\_LB\_MISREHSW<sub>n</sub> field descriptions**

| Field                        | Description                                                                                       |
|------------------------------|---------------------------------------------------------------------------------------------------|
| 0–31<br>MISRESW <sub>x</sub> | On-Line MISR Expected high Bits<br>This field defines the high part (63-32) of the Expected MISR. |

**56.9.39 STCU2 On-Line LBIST MISR Read Low Register (STCU2\_LB\_MISRRLSW<sub>n</sub>)****NOTE**

The reset value is chip-specific; see the chapter that describes how modules are configured and connected.

The STCU2\_LB\_MISRRLSW register report the LSB part (31-0) of the MISR obtained at the end of the On-Line LBIST controller execution.

The size of the register depends on the number of MISR bits of the related LBIST.

The content of this register is initialized to its reset value as soon as STCU2\_RUNSW[RUNSW] = 1.

Address: 0h base + 128h offset + (64d × i), where i=0d to 5d

|       |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0        | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | MISRRSWx |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*       | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

### STCU2\_LB\_MISRRLSWn field descriptions

| Field            | Description                                                                                                                                       |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>MISRRSWx | On-Line MISR Read Low Bin<br><br>This field is equivalent to the Low Bits (31-0) of the MISR obtained at the end of the Off-Line LBIST execution. |

## 56.9.40 STCU2 On-Line LBIST MISR Read High Register (STCU2\_LB\_MISRRHSWn)

### NOTE

The reset value is chip-specific; see the chapter that describes how modules are configured and connected.

The STCU2\_LB\_MISRRHSW register report the MSB part (63-32) of the MISR obtained at the end of each On-Line LBIST controller execution.

The size of the register depends on the number of MISR bits of the related LBIST.

The content of this register is initialized to its reset value as soon as STCU2\_RUNSW[RUNSW] = 1.

Address: 0h base + 12Ch offset + (64d × i), where i=0d to 5d

|       |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0        | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | MISRRSWx |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*       | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

## Register description

- x = Undefined at reset.

### STCU2\_LB\_MISRRHSWn field descriptions

| Field            | Description                                                                                                                                      |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>MISRRSWx | On-Line MISR Read High Bits<br>This field is equivalent to the Low Bits (63-32) of the MISR obtained at the end of the Off-Line LBIST execution. |

## 56.9.41 STCU2 MBIST Control Register (STCU2\_MB\_CTRLn)

### NOTE

The reset value is chip-specific; see the chapter that describes how modules are configured and connected.

The STCU2\_MB\_CTRL register defines the control setting of MBIST controller.

The R/W fields in this register are readable at any time. You can write to these fields when either of the following conditions is true:

- Off-line Self-Test phase is active
- On-line Self-Test phase is active and STCU2\_CFG[WRP] = 0

The offset of this register is a function of NMCUT .

Address: 0h base + 600h offset + (4d × i), where i=0d to 64d

|       |     |    |     |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-----|----|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0   | 1  | 2   | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | CSM |    | PTR |    |    |    |    |    | 0  |    |    |    |    |    |    |    |
| W     |     |    |     |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*  | x* | x*  | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |
| Bit   | 16  | 17 | 18  | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0   |    |     |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |     |    |     |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*  | x* | x*  | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |

\* Notes:

- x = Undefined at reset.

### STCU2\_MB\_CTRLn field descriptions

| Field    | Description                                                          |
|----------|----------------------------------------------------------------------|
| 0<br>CSM | Concurrent/sequential mode<br>0 Sequential mode<br>1 Concurrent mode |

Table continues on the next page...

**STCU2\_MB\_CTRLn field descriptions (continued)**

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1–7<br>PTR       | <p>Next LBIST or MBIST pointer</p> <p>PTR defines the logical pointer to the next LBIST or MBIST to be scheduled. The next LBIST or MBIST is scheduled concurrently to the current one if the CSM bit is set to 1; otherwise it is scheduled sequentially to the completion of the current MBIST execution. In case of NIL pointer the CSM bit must be set Sequential (0) to define this is the end of the list. The self-testing procedure is stopped when the current MBIST has been completed.</p> <p>0h to (LBIST -1): pointer to LBIST 10h to (10h + MBIST -1b): pointer to MBIST 7Fh: pointer to NIL. No BIST execution. others: invalid pointer =&gt; an error is set into the STCU2_ERR register.</p> |
| 8–31<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

**56.10 Use cases and limitations**

The STCU2 module is designed to give high flexibility during the on/off-line test. In the following are reported the programming operations that have to be performed to correctly use the functionality of this module considering the three typical usage cases:

- the Off-line Self Test Sequence after a reset trigger event
- the bypass mode of the Self Test Sequence after a reset trigger event
- the On-line Self-test Sequence

**56.10.1 Offline self-test sequence**

This is the typical mode of using the STCU2 module after reset trigger event is applied to STCU2. The SSCM DCF bus is used to retrieve the STCU2 schedule and L/MBIST execution parameters stored into the NVM memory. The target is to cover the amount of physical defects in the digital logic and in the System RAMs/ROMs according to the System specifications. The sequence is the following:

- Unlock the Off-Line STCU2 access writing the Key1/Key2 sequence into the STCU2\_SKC register
- Program the LBIST fault reaction conditions (UF/RF) setting the STCU2\_LBUFM register
- Program the MBIST fault reaction conditions (UF/RF) setting, depending on the number of memories, the STCU2\_MBUFML/M/H registers
- Program the STCU2 internal fault reaction conditions (UF/RF) setting the STCU2\_ERR\_FM register
- Program the STCU2\_LB\_CTRL, STCU2\_LB\_PCS, STCU2\_MISREL/H registers of each LBIST to be executed.
- Program the STCU2\_MB\_CTRL registers of each NMCUT to be executed

- Program the STCU2\_WDG register to define the time budget assigned for the L/MBIST execution
- Program the STCU2\_CFG register to define: the core and LBIST/MBIST TCK clock prescaling factor setting the CLK\_CFG bits, the LB\_DELAY field to delay the LBIST start when many LBIST are run concurrently to mitigate the potential transient power issue during the start-up phase, the PMOSEN or MBU in order to define the algorithm to be used during MBIST run, and finally set the pointer to the first LBIST/NMCUT to be executed
- In case the PLL usage has been enabled, program the STCU2\_PLL\_CFG value in order to set the required system frequency
- Program the RUN bit into the STCU2\_RUN register to enable the Off-Line Self-Test execution and the L/MBPLEN bits to enable the PLL during L/MBIST execution according to STCU2\_PLL\_CFG register content
- Start the Off-Line Self-Test and, depending on PLL management and L/MBIST execution order, when PLL LOCK is high.
- Wait the WDGEOC run time execution specified into the STCU2\_WDG register.
- The STCU2 switches off the core clock at the end of the Self-Test run and releases the PLL control in case this feature has been enabled.
- The STCU2 resets the STCU2\_RUN[RUN] bit and the device exits from boot sequence. In case there are failures, the STCU2 flags these failures toward the FCCU.
- The software application reads the STCU2\_LBS flag register to check the failed LBIST when UF or RF conditions is/are detected.
- The software application reads the STCU2\_LBE flag register to check the LBIST still on-going when UF or RF is/are detected.
- The software application reads the STCU2\_MBSL/M/H flag registers, depending on the number of memories, to check the failed RAMs/ROMs Memory BIST when UF or RF is/are detected.
- The software application reads the STCU2\_MBEL/M/H flag registers, depending on the number of memories, to check if the RAMs/ROMs Memory BIST is still on-going when UF or RF is/are detected.
- The software application reads the bits INV, ENGE, WDTO, LOCKE of the STCU2\_ERR\_STAT register to check whether there has been an internal STCU2 engine/parameters failure when UF or RF is/are detected.
- The software application reads for each device's LBIST, the STCU2\_LBMISREL/H and STCU2\_LBIST\_MISRRL/H registers to check the coherence with the STCU2\_LBS bits.



**NOTE**

The software application reads operations on the STCU2 and LBIST MISR registers after off-line self test execution has been successfully performed are just an additional reliability layer added to improve the already implemented Auto Self-Test feature included into STCU2.

**56.10.2 Online self test sequence**

The online self test sequence is the most commonly used STCU2 application mode. The IPS interface is used to program the STCU2 registers and to schedule LBIST and MBIST execution. Since online LBIST and MBIST operations clean-up the content of Digital Logic (LBIST) or RAM (MBIST) tested, activation of this mode must be approached with caution. The suggested sequence is the following:

1. Write the Key1/Key2 sequence to the SKC register to unlock online STCU2 access.
2. Ensure CFG[WRP]=0 to enable IPS bus access to the online self test registers.
3. If required, program the following registers to select the LBIST, MBIST and STCU2 internal fault reaction: LBUFM, MBUFML/M/H and ERR\_FM.

**NOTE**

The LBIST, MBIST and STCU2 internal fault reaction should have been already programmed during the offline self test sequence and so there should be no need to program again. However, in case they have not been programmed or there is the need of changing them for any reason, they can be overwritten.

4. Select the reset management options for online LBIST execution by programming the LBRMSW register.
5. The LB\_CTRL and LB\_PCS registers should have been already programmed during offline self test sequence and so they should not need to be programmed again. However, if they have not been programmed or there is the need of changing them for any reason, they can be overwritten. The registers LB\_MISRELSW/HSW have to be programmed according to the expected signature.
6. Overwrite/Program the MB\_CTRL registers of each NMCUT to be executed.
7. Overwrite/Program the WDG register to define the time budget assigned for the L/MBIST execution
8. Overwrite/Program the CFG register to define: the core and L/MBIST TCK clock prescaling factor setting the CLK\_CFG bits, the LB\_DELAY field to delay the LBIST start when many LBIST are run concurrently to mitigate the potential transient power issue during the start-up phase, the PMOSEN or MBU in order to

define the algorithm to be used during MBIST run, and finally set the pointer to the first LBIST/NMCUT to be executed.

9. Program the RUNSW[RUNSW] field to enable online self test execution; the L/MSWPLEN bits to enable the PLL lock signal monitor during L/MBIST execution and the MBIE to enable the related Interrupt requests.
10. Start the online self test and depending on PLL management and L/MBIST execution order, also when PLL LOCK is high.
11. Wait the WDGEOC run time execution specified into the WDG register
12. In case the MBIE have been enabled, an interrupt is generated at the end of all the MBIST execution. The software application has to manage these interrupt signals and clean-up the related flag (MBIFLG) into the INT\_FLG register.
13. The STCU2 switch-off the core clock at the end of the self test run
14. The functional reset activated is managed by the LBRMSW register (See step 4).
15. In case, at least one of the selected LBIST enables the Global functional reset, the STCU2 requests a functional reset which restarts the system while, when all the selected LBIST enables its own dedicated functional reset, only these local reset lines are applied and only the related LBIST partitions are cleaned-up.
16. The STCU2 resets the RUNSW[RUNSW] field. In both the reset cases and in case of fault/s, the STCU2 flags the failure toward the FCCU.
17. The software application reads the LBSSW flag register to check the failed LBIST when UF or RF conditions is/are detected.
18. The software application reads the LBESW flag register to check if the LBIST is still on-going when UF or RF is/are detected.
19. The software application reads the MBSLSW/MSW/HSW flag registers, depending on the number of memories, to check the failed RAMs/ROMs Memory BIST when UF or RF is/are detected.
20. The software application reads the MBELSW/MSW/HSW flag registers, depending on the number of memories, to check if the RAMs/ROMs Memory BIST is still on-going when UF or RF is/are detected.
21. The software application reads the bit INVPSW, ENGESW, WDTOSW, LOCKESW of the ERR\_STAT register to check whether there has been an internal STCU2 engine/parameters failure when UF or RF is/are detected.
22. The software application reads for each device's LBIST, the LBMISRELSW/HSW and LBIST\_MISRRLSW/HSW registers to check the coherence with the LBS bits

### NOTE

1. A reset is necessary after running any kind of online self test.
2. The software application read operations on the STCU2 and LBIST MISR registers after successful online self test execution are an additional reliability layer to improve the auto-self-test feature included in the STCU2.

3. Flash memory must be in the idle state when BIST execution is started. Program and erase functionality must not be enabled when entering BIST execution.

### 56.10.3 Bypass USER Mode

In this case, the USER application parameters stored in FLASH and read by SSCM are written in order to skip the Self-Test procedure after a reset trigger event is applied. It might be useful in case the device is not configured for applications requiring improved reliability. The sequence is the following:

1. Unlock the STCU2 access writing the Off-Line Key1/Key2 sequence into the STCU2\_SKC register
2. Set the BYP bit in the STCU2\_RUN register

After BYP bit is set, the core clock is switched off, the Self-Test sequence is not applied and the system can wake up and start the user application.

### 56.10.4 Design implementation information

The following list reports the limitations related to the current implementation:

- Every NMCUT BIST can be run once during the OnlineOffline self-test procedure.
- It is not possible to run LBIST and MBIST concurrently.
- Because software has full control of the system while applications are running during the online self-test, STCU2 is not allowed to take control of the PLL but can just monitor the lock signal to flag wrong unlock events.



## Chapter 57

# Cyclic Redundancy Check (CRC) Unit

### 57.1 Chip-specific CRC information

This chip has one instance of the CRC module. It supports three contexts:  $N = 3$ .

### 57.2 Introduction

The Cyclic Redundancy Check (CRC) computing unit is dedicated to the computation of CRC, off-loading the CPU. Each context has a separate CRC computation engine in order to allow the concurrent computation of the CRC of multiple data streams. The CRC computation is performed at speed without wait-state insertion. Bit-swap and bit-inversion operations can be applied on the final CRC signature. Each context can be configured with one of three hard-wired polynomials, normally used for most of the standard communication protocols. The data stream supports multiple data width (byte/halfword/word) formats.

### 57.3 Main features

- Three contexts for the concurrent CRC computation are supported
- Separate CRC engine for each context
- Zero-wait states during the CRC computation (pipeline scheme)
- Three hard-wired polynomials (CRC-8, CRC-32 ITU-T V.42, and CRC-16-CCITT)
- Support for byte, halfword, or word width of the input data stream

### 57.3.1 Standard features

- Peripheral bus interface
- CRC-8 VDA CAN
- CRC-16-CCITT
- CRC-32 ITU-T V.42 (see note below)

#### NOTE

Although the CRC-32 ITU-T V.42 polynomial is used for CRC generation, the generation algorithm differs (MISR) from the one used by the IEEE 802.3 standard.

## 57.4 Block diagram

The top level diagram of the CRC module is given in the following figure. Refer to [Main features](#) for number of contexts in this module.

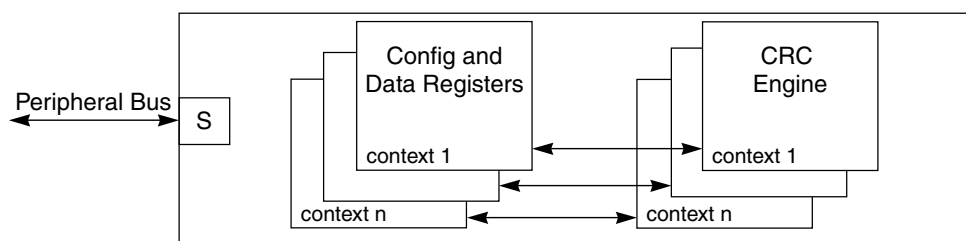


Figure 57-1. CRC top level diagram

## 57.5 External signal description

The CRC module does not generate any external signals.

### 57.5.1 Peripheral bus interface

The peripheral bus interface is a slave bus used for configuration and data streaming (CRC computation) purposes via CPU or DMA. The following bus operations (contiguous byte enables) are supported:

- Word (32 bits) data write/read operations to any registers

- Low and high halfword (16 bits, data[31:16] or data[15:0]) data write/read operations to any registers
- Byte (8 bits, data[31:24], data[23:16], data[15:8], or data[7:0]) data write/read operations to any registers
- Any other operation (free byte enables or other operations) must be avoided.

The CRC module generates a transfer error in the following cases:

- Any write/read access to the register addresses not mapped on the peripheral but included in the address space of the peripheral
- Any write/read operation different from byte/halfword/word (free byte enables or other operations) on each register

The registers of the CRC module are read/write accessible in each access mode:

- user
- supervisor

The following summarizes bus operation performance:

- Zero wait states (single bus cycle) for each write/read operation to the CRC\_CFG and CRC\_INP registers
- Zero wait states (single bus cycle) for each write operation to the CRC\_CSTAT register
- Double wait states (3 bus cycles) for each read operation to the CRC\_CSTAT or CRC\_OUTP registers immediately following (next clock cycle) a write operation to the CRC\_CSTAT, CRC\_INP, or CRC\_CFG registers belonging to the same context; in all the other cases, no wait states are inserted

## 57.6 CRC memory map and registers

**CRC memory map**

| Address offset (hex) | Register name                        | Width (in bits) | Access | Reset value                 | Section/page                |
|----------------------|--------------------------------------|-----------------|--------|-----------------------------|-----------------------------|
| 0                    | Configuration Register (CRC_CFG1)    | 32              | R/W    | <a href="#">See section</a> | <a href="#">57.6.1/2676</a> |
| 4                    | Input Register (CRC_INP1)            | 32              | R/W    | 0000_0000h                  | <a href="#">57.6.2/2677</a> |
| 8                    | Current Status Register (CRC_CSTAT1) | 32              | R/W    | FFFF_FFFFh                  | <a href="#">57.6.3/2678</a> |
| C                    | Output Register (CRC_OUTP1)          | 32              | R      | FFFF_FFFFh                  | <a href="#">57.6.4/2679</a> |

*Table continues on the next page...*

## CRC memory map (continued)

| Address offset (hex) | Register name                        | Width (in bits) | Access | Reset value                 | Section/ page               |
|----------------------|--------------------------------------|-----------------|--------|-----------------------------|-----------------------------|
| 10                   | Configuration Register (CRC_CFG2)    | 32              | R/W    | <a href="#">See section</a> | <a href="#">57.6.1/2676</a> |
| 14                   | Input Register (CRC_INP2)            | 32              | R/W    | 0000_0000h                  | <a href="#">57.6.2/2677</a> |
| 18                   | Current Status Register (CRC_CSTAT2) | 32              | R/W    | FFFF_FFFFh                  | <a href="#">57.6.3/2678</a> |
| 1C                   | Output Register (CRC_OUTP2)          | 32              | R      | FFFF_FFFFh                  | <a href="#">57.6.4/2679</a> |
| 20                   | Configuration Register (CRC_CFG3)    | 32              | R/W    | <a href="#">See section</a> | <a href="#">57.6.1/2676</a> |
| 24                   | Input Register (CRC_INP3)            | 32              | R/W    | 0000_0000h                  | <a href="#">57.6.2/2677</a> |
| 28                   | Current Status Register (CRC_CSTAT3) | 32              | R/W    | FFFF_FFFFh                  | <a href="#">57.6.3/2678</a> |
| 2C                   | Output Register (CRC_OUTP3)          | 32              | R      | FFFF_FFFFh                  | <a href="#">57.6.4/2679</a> |

## 57.6.1 Configuration Register (CRC\_CFGn)

Access: User read/write.

Address: 0h base + 0h offset + (16d × i), where i=0d to 2d

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | 0 |   |   |   |   |   |   |   | 0 |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       |    |    |    |    |    |    |    |    |    |    |               |    |              |    |       |    |      |     |
|-------|----|----|----|----|----|----|----|----|----|----|---------------|----|--------------|----|-------|----|------|-----|
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26            | 27 | 28           | 29 | 30    | 31 |      |     |
| R     | 0  |    |    |    |    |    |    |    |    |    | SWAP_BYTEWISE |    | SWAP_BITWISE |    | POLYG |    | SWAP | INV |
| W     |    |    |    |    |    |    |    |    |    |    |               |    |              |    |       |    |      |     |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0             | 0  | *            | *  | 0     | 0  |      |     |

\* Notes:

- POLYG field: Reset value is a function of n in CRC\_CFGn. When n%2 = 0, reset value = 01b. When n%2 = 1, reset value = 00b. So CRC\_CFG1 POLYG reset value = 00b (Since 1%2=1).

## CRC\_CFGn field descriptions

| Field           | Description                                                                             |
|-----------------|-----------------------------------------------------------------------------------------|
| 0–7<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |

Table continues on the next page...



**CRC\_CFGn field descriptions (continued)**

| Field                   | Description                                                                                                                                                                                                                                                                                        |
|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 8–25<br>Reserved        | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                            |
| 26<br>SWAP_<br>BYTEWISE | Swap CRC_INP byte-wise<br><br><b>NOTE:</b> INV and SWAP bits are set to 1 for CRC-32 polynomial calculations.<br><br>0 Do not swap<br>1 Perform byte-wise swap on CRC_INP input data internally for CRC-32 polynomial calculations.                                                                |
| 27<br>SWAP_BITWISE      | Swap CRC_INP bit-wise<br><br>0 Do not swap<br>1 Perform bit-wise swap on CRC_INP input data internally for CRC-8 and CRC-16 polynomial calculations.                                                                                                                                               |
| 28–29<br>POLYG          | Polynomial selection<br><br>This bit can be written only during the configuration phase.<br><br>00 CRC-CCITT polynomial<br>01 CRC-32 polynomial<br>10 CRC-8 polynomial<br>11 Reserved                                                                                                              |
| 30<br>SWAP              | Swap selection<br><br>This bit can be written only during the configuration phase. The swap operation is a bit-by-bit swapping of the content.<br><br>0 No swap selection applied on the CRC_OUTP content<br>1 Swap selection (MSB to LSB, LSB to MSB) applied on the CRC_OUTP content.            |
| 31<br>INV               | Inversion selection<br><br>This bit can be written only during the configuration phase. The inversion operation is a complement (or negation) of the content.<br><br>0 No inversion selection applied on the CRC_OUTP content<br>1 Inversion selection (bit x bit) applied on the CRC_OUTP content |

**57.6.2 Input Register (CRC\_INPn)**

Access: User read/write.

**NOTE**

High level code must be written in a way to realize byte-, halfword- and word-wide writes on assembler code level. Any endianness problem faced can be corrected by appropriate configuration of CRC\_CFG[SWAP\_BYTEWISE] and CRC\_CFG[SWAP\_BITWISE].

## CRC memory map and registers

Address: 0h base + 4h offset + (16d × i), where i=0d to 2d

|       |     |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-----|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0   | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | INP |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |     |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0   | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

### CRC\_INP*n* field descriptions

| Field       | Description                                                                                                                                                                                                                                                          |
|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>INP | Input data for the CRC computation<br><br>This register can be written with word-, halfword- (high or low), or byte-wide writes in any sequence. Only the bits written are fed to the CRC engine. In case of halfword write operation, the bytes must be contiguous. |

## 57.6.3 Current Status Register (CRC\_CSTAT*n*)

Access: User read/write.

Address: 0h base + 8h offset + (16d × i), where i=0d to 2d

|       |       |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0     | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | CSTAT |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |       |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 1     | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  |    |

### CRC\_CSTAT*n* field descriptions

| Field         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>CSTAT | CRC signature status<br><br>This register includes the current status of the CRC signature. No bit swap and inversion are applied to this register. In the case of the CRC-CCITT polynomial, only the 16 least-significant bits are significant. The 16 most-significant bits are set to 0 during the computation. In the case of the CRC-8 polynomial, only the 8 least-significant bits are significant. The 24 most-significant bits are set to 0 during the computation. The CSTAT register can be written at byte, halfword, or word. This register can be written only during the configuration phase. |

## 57.6.4 Output Register (CRC\_OUTP<sub>n</sub>)

Access: User read-only.

Address: 0h base + Ch offset + (16d × i), where i=0d to 2d

|       |      |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | OUTP |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |      |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 1    | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  |    |

**CRC\_OUTP<sub>n</sub> field descriptions**

| Field        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>OUTP | <p>Final CRC signature</p> <p>This register includes the final signature corresponding to the CRC_CSTAT register value eventually swapped and inverted. In the case of the CRC-CCITT polynomial, only the 16 least-significant bits are significant. The 16 MSB bits are set to 0 during the computation. In the case of the CRC-8 polynomial, only the 8 least-significant bits are significant. The 24 most-significant bits are set to 0 during the computation.</p> |

## 57.7 Functional description

The CRC module supports the CRC computation for each context. Each context has its own complete set of registers, including the CRC engine. The data flow of each context can be interleaved. The data stream can be structured as a sequence of bytes, halfwords, or words. The input data sequence is provided, eventually mixing the data formats (byte, halfword, and word), writing to the input data register (CRC\_INP).

The data stream is generally executed by *n* concurrent DMA data transfers (mem2mem) where *n* is less or equal to the number of contexts.

The standard generator polynomials are given in [Equation 2 on page 2679](#), [Equation 3 on page 2679](#), and [Equation 4 on page 2680](#) for the CRC computation of each context. Two separate registers are available: CRC\_INP for writing data and CRC\_CSTAT for retrieving the (accumulated up to now) CRC.

$$x^8 + x^4 + x^3 + x^2 + 1$$

**Equation 2. CRC8 VDA CAN**

$$x^{16} + x^{12} + x^5 + 1$$

**Equation 3. CRC-CCITT (x.25 protocol)**

$$x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^8+x^7+x^5+x^4+x^2+x+1$$

### Equation 4. CRC-32 (ITU-T V.42 protocol)

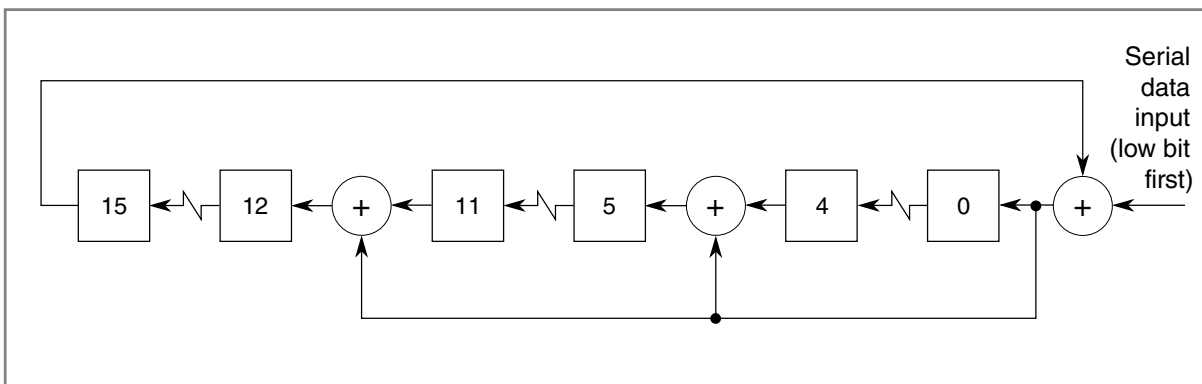


**Figure 57-18. CRC-CCITT engine concept scheme**

The initial seed value of the CRC can be programmed by initializing the CRC\_CSTAT register. A diagram illustrating serial data loading of the CRC engine is shown in [Figure 57-18](#) for the CRC-CCITT. Note that, conceptually, the least-significant, or "right-most," bit shown in [Input Register \(CRC\\_INP<sub>n</sub>\)](#), is fed first into the engine. The actual implementation executes the CRC computation in a single clock cycle (parallel data loading). A pipeline scheme has been adopted to decouple the IPS bus interface from the CRC engine in order to allow the computation of the CRC at speed (zero wait states).

If the CRC signature is used for encapsulation in the data frame of a communication protocol (for example, SPI, etc.), a bit swap (high bit for low bit or low bit for high bit) and/or bit inversion of the final CRC signature can be applied to CRC\_OUTP before transmitting the CRC.

Use of the CRC module is summarized in the flow chart given in [Figure 57-19](#).

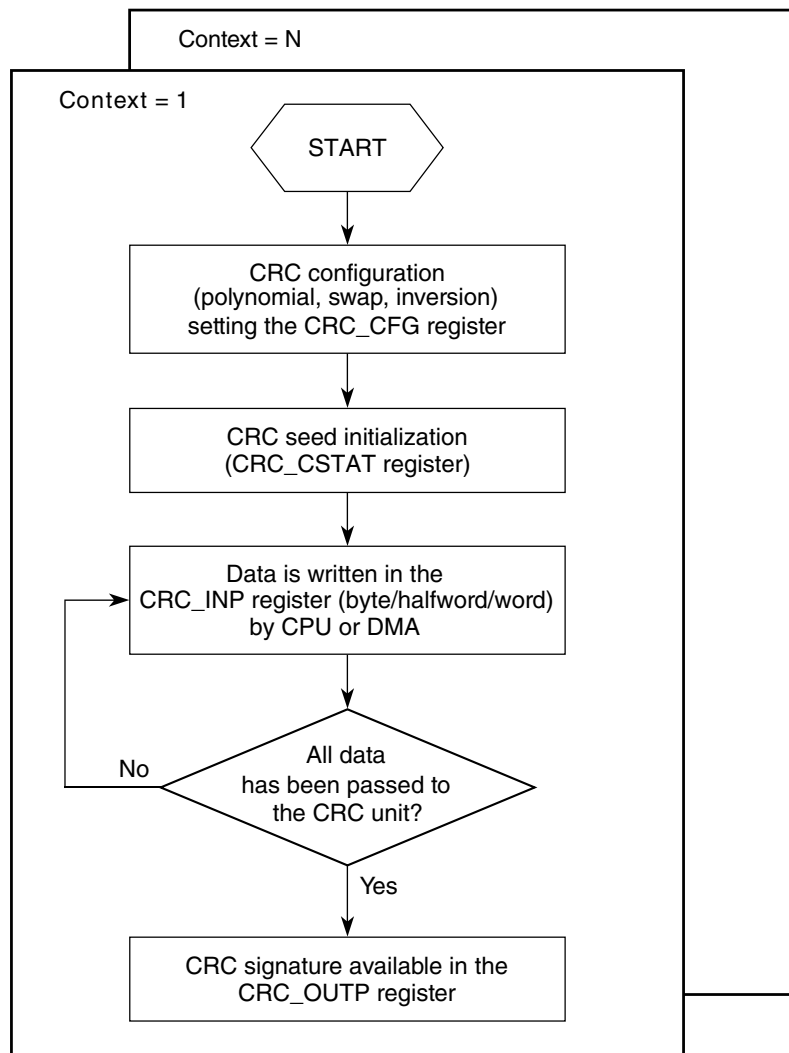


Figure 57-19. CRC computation flow

## 57.8 Use cases

### 57.8.1 Programming example

The number of contexts depends on the application. The overall number of contexts for the CRC peripheral depends on the number of peripherals that concurrently require intervention by the CRC module. Two main use cases are considered:

- Calculation of the CRC of the configuration registers during the process safety time
- Calculation of the CRC on the incoming/outcoming frames for the communication protocols (not protected with CRC by definition of the protocol itself) used as a safety-relevant peripheral

The signature of the configuration registers is computed correctly only if these registers do not contain any status bit.

Assuming that the DMA engine has  $n$  channels (greater than or equal to the number of contexts) configurable for the following types of data transfer—mem2mem, periph2mem, mem2periph—the following sequence, as shown in the following figure, is applied to manage the transmission data flow:

1. DMA/CRC module configuration (context  $x$ , channel  $x$ ) by CPU
2. Payload transfer from the MEM to the CRC module (CRC\_INP register) after MSB-to-LSB change (input mirroring) to calculate the CRC signature (phase1) by DMA (mem2mem data transfer, channel  $x$ )
3. CRC signature copy from the CRC module (CRC\_OUTP register) to the MEM (phase 2) by CPU
4. Data block (payload + CRC) transfer from the MEM to the PERIPH module (for example, SPI Tx FIFO) (phase 3) by DMA (mem2periph data transfer, channel  $x$ )

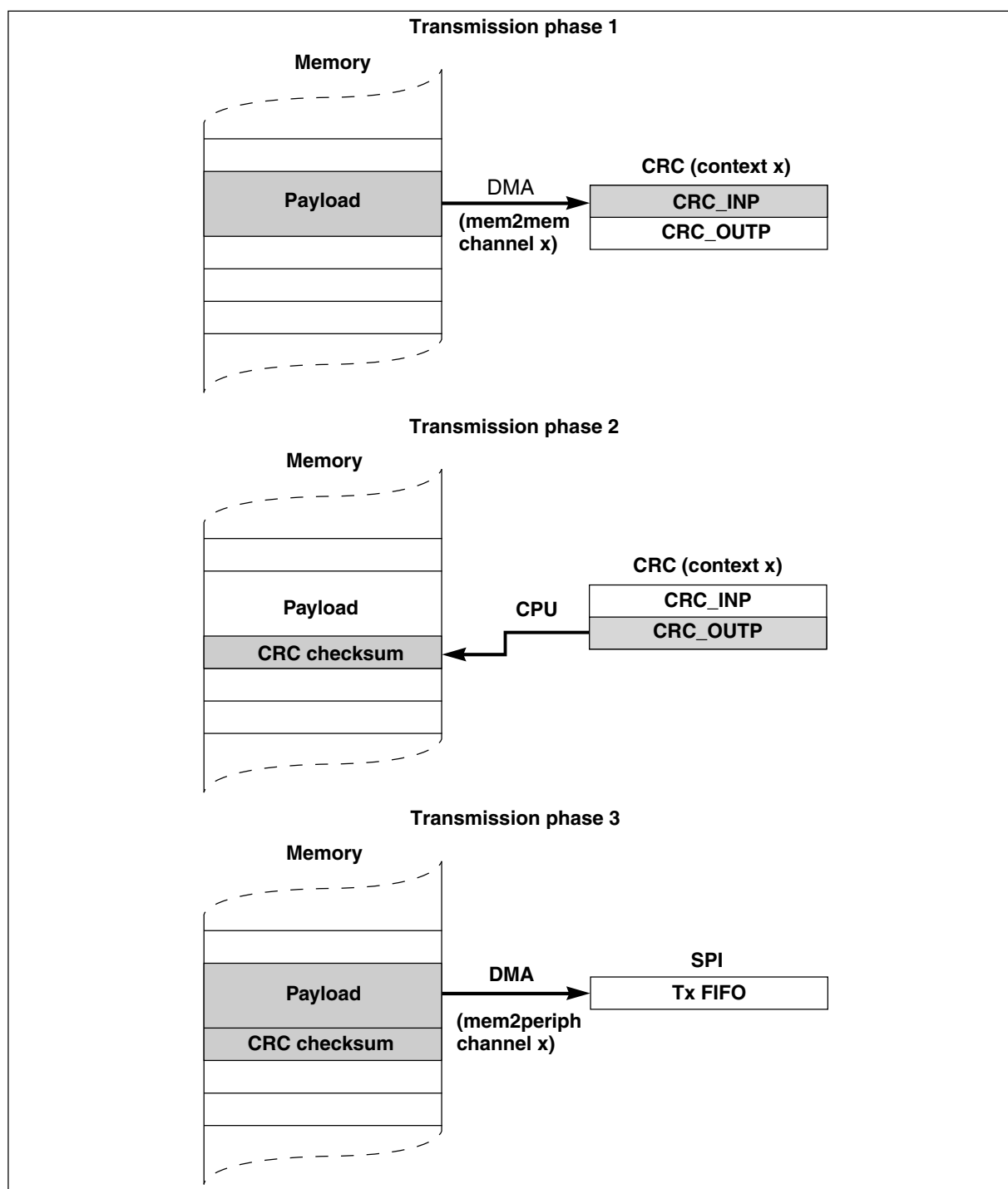


Figure 57-20. DMA-CRC transmission sequence

## 57.8.2 Register programming

- INV and/or SWAP (affecting the output only) can be applied to CRC-8, CRC-16, and CRC-32. Both can be applied together.

#### Use cases

- SWAP\_BITWISE (affecting the input only) can be applied for all polynomials if required by an application.
- SWAP\_BYTEWISE (affecting the input only) can be applied for CRC-16 and CRC-32 polynomials only.
- SWAP\_BITWISE has priority over SWAP\_BYTEWISE when both are applied together.
- When generating CRC-32 for the ITU-T V.42 standard the user needs to set SWAP\_BYTEWISE together with INV and SWAP.
- When generating CRC-16 the user needs to set SWAP\_BITWISE bit.



# Chapter 58

## Security Overview

### 58.1 Introduction

This chip has a comprehensive set of customer-configurable security features designed to protect code and data from unauthorized access.

### 58.2 Security features summary

The chip's security features include:

- Device censorship based on life cycle model
- Memory security:
  - Password protection for flash memory with PASS module
  - One-time-programmable (OTP) flash memory areas
  - Tamper detection including flash memory erase counter
  - MBIST functionality that can be enabled for execution during an STCU2 off-line condition to initialize RAMs (see the STCU2 chapter for details about MBIST execution in an STCU2 off-line condition)
  - BAM with software routines to initialize RAMs when not booting from internal flash memory
  - Security functions described in the *Secure Hardware Extension (SHE) Functional Specification Version 1.1*
- Monitoring of operating conditions
- Security watchdog
- Secure debug

## 58.3 Flash memory read, write, and erase protection

[Flash memory read protection scenarios](#) explains the supported options for read protection.

The C55FMC and PASS modules provide write and erase protection:

- The C55FMC\_LOCK $n$  registers provide one level of write and erase protection.
- The PASS\_LOCK $n$ \_PG $m$  registers provide a higher level of password-protected write and erase protection that supersedes C55FMC's protection by overwriting the values of the C55FMC\_LOCK $n$  registers.
- After bootup, during which the values of PASS\_LOCK $n$ \_PG $m$  registers are loaded from DCF records: Users can override these values temporarily, until the next reset, by programming them to different values, thereby also temporarily programming different values for the C55FMC\_LOCK $n$  registers.

For details about these registers, see the [Embedded Flash Memory \(C55FMC\)](#) and [Password and Device Security Module \(PASS\)](#) chapters.

# Chapter 59

## Cryptographic Services Engine (CSE)

### 59.1 Chip-specific CSE information

This chip has one instance of the CSE module. The module:

- executes the chip's secure boot process. See [Parallel secure boot](#) in the System Boot details.
- has exclusive access to the flash memory blocks mapped to the C55FMC\_LOCK1 register, PASS\_LOCK1\_PGn registers, and TDRn\_LOCK1 DCF client. See [C55FMC\\_LOCK1 register bit mapping](#). The system MPU is automatically configured to prevent other bus masters from interfering with CSE's access to the flash memory.

#### 59.1.1 CSE configuration and SECURE\_BOOT execution

The [UTEST Secure Boot Configurations DCF client](#) automatically configures aspects of CSE operation at startup.

See this DCF client's description for details about its factory-programmed value, which ensures that reset logic issues the SECURE\_BOOT command to CSE during reset.

#### 59.1.2 CSE Clk Freq

In the description of CSE\_CR[DIV], the mentioned *CSE Clk Freq* corresponds to the FM\_PER\_CLK frequency on this chip. See the Clocking chapter for more information about FM\_PER\_CLK.

### 59.1.3 CSE clock: minimum frequency

To ensure compliance with SHE specification performance requirements, the CSE clock source (FM\_PER\_CLK) must have a minimum frequency of 105 MHz.

### 59.1.4 CSE debug not supported

This chip has no debug capability for the CSE module. Descriptions of CSE debug features in this chapter do not apply.

### 59.1.5 CSE command-processing completion before Stop mode entry

To ensure proper CSE operation: Before setting SIU\_HLT1[CSE] to 1, ensure CSE\_SR[BSY] is 0.

## 59.2 Introduction

### 59.2.1 Overview

The Cryptographic Services Engine (CSE) is a peripheral module that implements the security functions described in the *Secure Hardware Extension (SHE) Functional Specification Version 1.1*.

The CSE design includes a host interface with a set of memory mapped registers that are used by the CPU to issue commands and a system bus interface that allows the CSE to directly access system memory. Two dedicated blocks of system flash memory are used by the CSE for secure key storage.

### 59.2.2 Features

The CSE has the following features:

- Secure storage for cryptographic keys
- AES-128 encryption and decryption
- AES-128 CMAC authentication

- True random number generation
- Secure boot mode
- System bus master interface

### 59.2.3 Modes of operation

The CSE supports operation in Normal and Debug modes of operation. The use of the cryptographic keys stored by the CSE is controlled based on the activation of the CPU debug port and the successful completion of the secure boot process.

The CSE has a low-power mode that disables the clock to all logic except the host interface. Register accesses are supported in this mode, but commands are not processed.

### 59.2.4 Block diagram

The CSE design includes a command processor, host interface, system bus interface, local memory, AES logic, and True Random Number Generator (TRNG) as shown below.

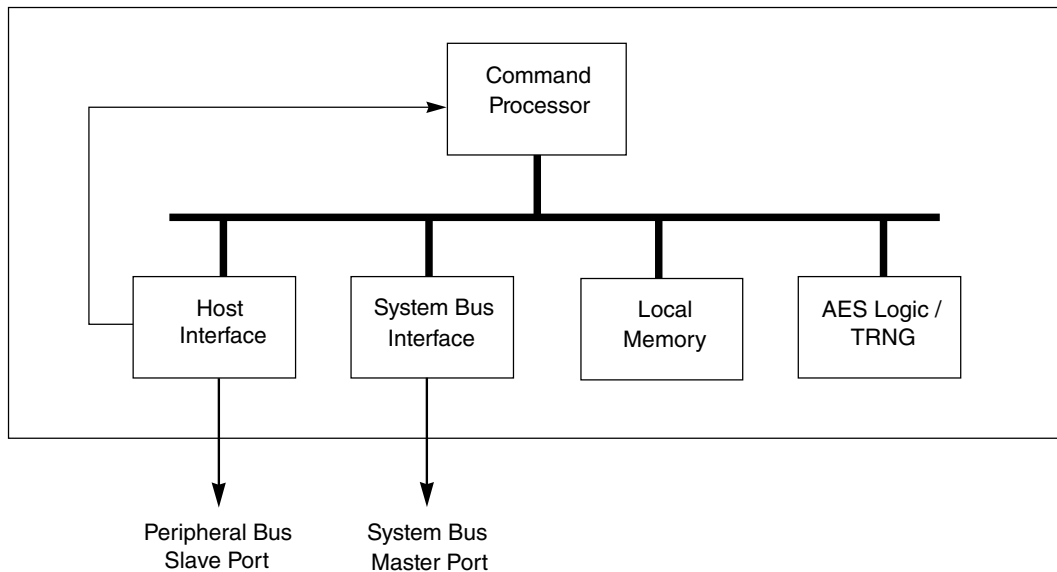


**Figure 59-1. CSE block diagram**

## 59.3 External signal description

The CSE has no external interface signals.

## 59.4 Memory Map and Register Definition

The CSE programming model has ten 32-bit registers. The programming model can only be accessed using 32-bit (word) accesses. References using a different size are invalid. The CSE memory map is shown in the table below.

**CSE memory map**

| Address offset (hex) | Register name                     | Width (in bits) | Access | Reset value                 | Section/page                |
|----------------------|-----------------------------------|-----------------|--------|-----------------------------|-----------------------------|
| 0                    | CSE Control Register (CSE_CR)     | 32              | R/W    | <a href="#">See section</a> | <a href="#">59.4.1/2690</a> |
| 4                    | CSE Status Register (CSE_SR)      | 32              | R      | 0000_0000h                  | <a href="#">59.4.2/2692</a> |
| 8                    | CSE Interrupt Register (CSE_IR)   | 32              | w1c    | <a href="#">See section</a> | <a href="#">59.4.3/2693</a> |
| C                    | CSE Error Code Register (CSE_ECR) | 32              | R      | 0000_0000h                  | <a href="#">59.4.4/2694</a> |
| 20                   | CSE Command Register (CSE_CMD)    | 32              | R/W    | 0000_0000h                  | <a href="#">59.4.5/2695</a> |
| 24                   | CSE Parameter Register (CSE_P1)   | 32              | R/W    | 0000_0000h                  | <a href="#">59.4.6/2696</a> |
| 28                   | CSE Parameter Register (CSE_P2)   | 32              | R/W    | 0000_0000h                  | <a href="#">59.4.6/2696</a> |
| 2C                   | CSE Parameter Register (CSE_P3)   | 32              | R/W    | 0000_0000h                  | <a href="#">59.4.6/2696</a> |
| 30                   | CSE Parameter Register (CSE_P4)   | 32              | R/W    | 0000_0000h                  | <a href="#">59.4.6/2696</a> |
| 34                   | CSE Parameter Register (CSE_P5)   | 32              | R/W    | 0000_0000h                  | <a href="#">59.4.6/2696</a> |

### 59.4.1 CSE Control Register (CSE\_CR)

The CSE\_CR contains fields for configuring and controlling operation of the CSE.

Address: 0h base + 0h offset = 0h

|       |     |    |    |    |    |    |    |    |    |     |    |     |      |     |     |     |
|-------|-----|----|----|----|----|----|----|----|----|-----|----|-----|------|-----|-----|-----|
| Bit   | 0   | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9   | 10 | 11  | 12   | 13  | 14  | 15  |
| R     | 0   |    |    |    |    |    |    |    |    |     |    |     |      |     |     |     |
| W     |     |    |    |    |    |    |    |    |    |     |    |     |      |     |     |     |
| Reset | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0  | 0   | 0    | 0   | 0   | 0   |
| Bit   | 16  | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25  | 26 | 27  | 28   | 29  | 30  | 31  |
| R     | DIV |    |    |    |    |    |    |    | 0  | KBS |    | SFE | MDIS | SUS | DRE | CIE |
| W     |     |    |    |    |    |    |    |    |    |     |    |     |      |     |     |     |
| Reset | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0  | 0   | 0    | 0   | 0   | *   |

\* Notes:

- CIE field: The reset value of this bit is 0 on POR, but when the reset lifts and CSE execution starts, the value of this bit becomes 1. So, after the first read access to this bit, the read value is 1.

## CSE\_CR field descriptions

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                             |
| 16–23<br>DIV      | TRNG Clock Divider Select. Sets the clock divide ratio for the TRNG clock. The TRNG clock is the module clock divided by a programmable ratio:<br>$\text{TRNG Clk Freq} = \text{CSE Clk Freq} / (2 * (\text{DIV} + 1))$ The divide ratio must be set such that the TRNG clock frequency is between 500 kHz and 2MHz.<br>0x00 divide by 2<br>0x01 divide by 4<br>0x02 divide by 6<br>...<br>0xFE divide by 510<br>0xFF divide by 512 |
| 24–25<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                             |
| 26<br>KBS         | Key Bank Select. Selects which set of application keys are visible for commands. See <a href="#">Table 59-13</a> .<br>0 Select KEY_1-KEY_10<br>1 Select KEY_11-KEY_20                                                                                                                                                                                                                                                               |
| 27<br>SFE         | Security Flag Extension. When the SFE bit is set, the CSE recognizes the VERIFY_ONLY security flag during LOAD_KEY processing; otherwise, it is ignored. The SFE bit has no effect on the enforcement of the VERIFY_ONLY security flag once a key is loaded.<br>0 Security Flag Extension disabled.<br>1 Security Flag Extension enabled.                                                                                           |
| 28<br>MDIS        | Module Disable. When the MDIS bit is set, the CSE is put into a low-power mode that disables the clock to all of the CSE logic except for the host interface. The MDIS bit should not be set during command processing (CSE_SR[BSY] = 1). The current command should be canceled and processing stopped (CSE_SR[BSY] = 0) before setting the MDIS bit.<br>0 Normal mode<br>1 Low-power mode                                         |
| 29<br>SUS         | Suspend Command Processing. When the SUS bit is set, the CSE suspends processing of the current command until the SUS bit is cleared. The current execution status of the command processor is reflected by the CSE_SR[EX] flag.<br>0 Enable processing of commands.<br>1 Suspend command processing.                                                                                                                               |
| 30<br>DRE         | DMA Request Enable. When the DRE bit is set, the DMA request is asserted when command processing is completed.<br><b>NOTE:</b> DMA is not implemented on this chip.<br>0 DMA request not enabled.<br>1 DMA request enabled.                                                                                                                                                                                                         |
| 31<br>CIE         | Command Complete Interrupt Enable. When the CIE bit is set, an interrupt request is generated and the CSE_IR[CIF] flag is set when command processing is completed.<br>0 Command Complete Interrupt disabled.<br>1 Command Complete Interrupt enabled.                                                                                                                                                                              |

## 59.4.2 CSE Status Register (CSE\_SR)

The CSE\_SR contains flags indicating the status of the CSE. Reading CSE\_SR[24:31] is the same as the SHE GET\_STATUS command.

Address: 0h base + 4h offset = 4h

|       |    |    |    |    |    |    |    |    |    |     |     |     |     |     |     |    |     |
|-------|----|----|----|----|----|----|----|----|----|-----|-----|-----|-----|-----|-----|----|-----|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9   | 10  | 11  | 12  | 13  | 14  | 15 |     |
| R     | 0  |    |    |    |    |    |    |    |    |     |     |     |     |     |     |    |     |
| W     |    |    |    |    |    |    |    |    |    |     |     |     |     |     |     |    |     |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0   | 0   | 0   | 0   | 0   | 0  |     |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25  | 26  | 27  | 28  | 29  | 30  | 31 |     |
| R     | 0  |    |    |    |    |    |    |    | EX | IDB | EDB | RIN | BOK | BFN | BIN | SB | BSY |
| W     |    |    |    |    |    |    |    |    |    |     |     |     |     |     |     |    |     |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0   | 0   | 0   | 0   | 0   | 0  |     |

### CSE\_SR field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                             |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–22<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                 |
| 23<br>EX         | Execute.<br><br><b>NOTE:</b> Ignore the status of CSE_SR[EX] when CSE_SR[BSY] flag is '0'.<br><br>0 Command processor is idle or suspended.<br>1 Command processor is executing a command.                                                                                                              |
| 24<br>IDB        | Internal Debug. The IDB bit is set by the SECURE_BOOT, INIT_CSE, and DEBUG_AUTH commands when no user keys (i.e. all keys except SECRET_KEY and UID) are stored. It is cleared on reset and by the LOAD_KEY command.<br><br>0 Internal Debug functions disabled.<br>1 Internal Debug functions enabled. |
| 25<br>EDB        | External Debug. The EDB bit is set when the CPU debug port is activated and is cleared on reset.<br><br>0 External debugger not attached.<br>1 External debugger attached.                                                                                                                              |
| 26<br>RIN        | Random Number Generator Initialized. The RIN bit is set by the INIT_RNG command and is cleared on reset and by the DEBUG_AUTH command.<br><br>0 Random number generator not initialized.<br>1 Random number generator initialized.                                                                      |
| 27<br>BOK        | Secure Boot OK. The BOK bit is set by the successful completion of the SECURE_BOOT command. It is cleared by reset and by the BOOT_FAILURE command.                                                                                                                                                     |

Table continues on the next page...



**CSE\_SR field descriptions (continued)**

| Field     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|           | 0 Secure boot not completed or secure boot failure.<br>1 Secure boot successful.                                                                                                                                                                                                                                                                                                                                                                                                      |
| 28<br>BFN | Secure Boot Finished. The BFN bit is set by the SECURE_BOOT command when the BIN bit is set, when an error is encountered, or when the BOOT_MAC value does not match. It is also set by the BOOT_OK or BOOT_FAILURE commands. It is cleared on reset. The BFN bit is not set in the event the CSE flash blocks are accessible, but contain an invalid firmware image as detected by a checksum mismatch (CSR_ECR=0x13).<br><br>0 Secure boot not finished.<br>1 Secure boot finished. |
| 29<br>BIN | Secure Boot Initialization. The BIN bit is set by the SECURE_BOOT command if the BOOT_MAC memory slot is empty and is cleared on reset.<br><br>0 Secure boot personalization not completed.<br>1 Secure boot personalization completed.                                                                                                                                                                                                                                               |
| 30<br>SB  | Secure Boot. The SB bit is set by the SECURE_BOOT command if the BOOT_MAC_KEY slot is not empty, and is cleared on reset.<br><br>0 Secure boot not activated.<br>1 Secure boot activated.                                                                                                                                                                                                                                                                                             |
| 31<br>BSY | Busy. The BSY bit is set when a command is issued and cleared when command processing is completed.<br><br>0 Command processing completed.<br>1 Command processing not completed.                                                                                                                                                                                                                                                                                                     |

**59.4.3 CSE Interrupt Register (CSE\_IR)**

The CSE\_IR contains the command completion interrupt flag bit.

Address: 0h base + 8h offset = 8h

|       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|
| Bit   | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15  |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   |
| Bit   | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31  |
| R     | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    | CIF |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | w1c |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | *   |

\* Notes:

- CIF field: The reset value of this bit is 0 on POR, but when the device boots in a mode that supports SECURE\_BOOT, the value of this bit becomes 1 upon the command's completion.

## CSE\_IR field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                             |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–30<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                 |
| 31<br>CIF        | Command Complete Interrupt Flag. The CIF flag reflects the state of the command complete interrupt request. The CIF flag and interrupt request are cleared by writing a 1 to this bit. Writing a 0 has no effect.<br><br>0 No interrupt request.<br>1 Interrupt request due to completion of a command. |

## 59.4.4 CSE Error Code Register (CSE\_ECR)

The CSE\_ECR contains the error code from the last completed command.

Address: 0h base + Ch offset = Ch

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | EC |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

## CSE\_ECR field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                               |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–26<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                   |
| 27–31<br>EC      | Error Code from the last command completed.<br>0x00 No error<br>0x02 Command sequence error<br>0x03 Key not available<br>0x04 Invalid key<br>0x05 Empty key<br>0x06 No secure boot<br>0x07 Key write protected<br>0x08 Key update error<br>0x09 Random number seed not initialized<br>0x0A Internal debug not allowed<br>0x0B Command issued while busy<br>0x0C System memory error<br>0x10 Internal memory error<br>0x11 Invalid command |

Table continues on the next page...

**CSE\_ECR field descriptions (continued)**

| Field | Description                           |
|-------|---------------------------------------|
|       | 0x12 TRNG error                       |
|       | 0x13 CSE flash block error            |
|       | 0x14 Internal command processor error |
|       | 0x15 Length error                     |

**59.4.5 CSE Command Register (CSE\_CMD)**

Commands are issued by first loading the appropriate parameter registers (CSE\_Px) and then writing a command code to the command register (CSE\_CMD). Reads of the CSE\_CMD register have no effect on the operation of the CSE. See [Command processing](#) for a description of command processing and [CSE commands](#) for a description of each command.

Address: 0h base + 20h offset = 20h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26  | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | CMD |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0   | 0  | 0  | 0  | 0  | 0  |

**CSE\_CMD field descriptions**

| Field            | Description                                                                                                                                                                                                                                                                                                            |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–26<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                |
| 27–31<br>CMD     | Command. See <a href="#">CSE commands</a> for details of each command.<br>0x01 = ENC_ECB<br>0x02 = ENC_CBC<br>0x03 = DEC_ECB<br>0x04 = DEC_CBC<br>0x05 = GENERATE_MAC<br>0x06 = VERIFY_MAC<br>0x07 = LOAD_KEY<br>0x08 = LOAD_PLAIN_KEY<br>0x09 = EXPORT_RAM_KEY<br>0x0A = INIT_RNG<br>0x0B = EXTEND_SEED<br>0x0C = RND |

*Table continues on the next page...*

**CSE\_CMD field descriptions (continued)**

| Field | Description         |
|-------|---------------------|
|       | 0x0D = SECURE_BOOT  |
|       | 0x0E = BOOT_FAILURE |
|       | 0x0F = BOOT_OK      |
|       | 0x10 = GET_ID       |
|       | 0x11 = CANCEL       |
|       | 0x12 = DEBUG_CHAL   |
|       | 0x13 = DEBUG_AUTH   |
|       | 0x14 = TRNG_RND     |
|       | 0x15 = INIT_CSE     |
|       | 0x16 = MP_COMPRESS  |

**59.4.6 CSE Parameter Register (CSE\_Pn)**

The five parameter registers (CSE\_Px) are used to hold command parameter values. See [Command processing](#) for a description of how to issue commands and [CSE commands](#) for a description of each command. The parameter registers are read only when the CSE\_SR[BSY] bit is set.

Address: 0h base + 24h offset + (4d × i), where i=0d to 4d

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**CSE\_Pn field descriptions**

| Field        | Description                                              |
|--------------|----------------------------------------------------------|
| 0–31<br>PARM | Command parameter (data value or address of data value). |

**59.5 CSE functional description**

The CSE implements a comprehensive set of cryptographic functions as described in the *SHE Functional Specification*, including secure key storage, AES encryption, secure boot, AES CMAC authentication, and random number generation.

## 59.5.1 Host interface

The host interface includes all of the CSE memory mapped registers (see the memory map and register section).

The Control Register (CSE\_CR) is used to set configuration options and control operation of the CSE. If the CSE\_CR[CIE] bit is set, an interrupt request is generated after the CSE finishes processing each command. The interrupt is cleared by writing a 1 to the CSE\_IR[CIF] bit. When the CSE\_CR[SUS] bit is set, the CSE suspends processing of the current command and does not respond to new commands including the CANCEL command.

Command processing is resumed by clearing the CSE\_CR[SUS] bit. The CSE cannot respond immediately to changes in the CSE\_CR[SUS] bit, so the CSE\_SR[EX] flag indicates the current status of the CSE command processor. The CSE\_SR[EX] flag is set when the CSE command processor is running and is cleared when the CSE is idle or has suspended processing. The CSE\_CR[MDIS] bit puts the CSE into a low-power mode. In low-power mode, registers can be accessed but commands are not processed. Before setting the CSE\_CR[MDIS] bit or entering any device low-power modes, application software should cancel any pending command and wait for the CSE\_SR[BSY] bit to be cleared. The CSE\_CR[SFE] bit enables loading of the VERIFY\_ONLY flag during the LOAD\_KEY command. The CSE\_CR[KBS] bit selects which key bank (KEY\_1–KEY\_10 or KEY\_11–KEY\_20) is active.

The Status Register (CSE\_SR) contains a set of nine status flags. Reading CSE\_SR[24:31] is equivalent to the SHE GET\_STATUS command. The CSE\_SR[BSY] bit indicates when the CSE is processing a command. The flag is cleared when processing is completed. Polling the CSE\_SR[BSY] bit is an alternative to interrupt-driven operation. The CSE\_SR[SB], CSE\_SR[BIN], CSE\_SR[BFN] and CSE\_SR[BOK] bits reflect the secure boot status (see [Secure boot](#)). The CSE\_SR[RIN] bit is set when the PRNG is initialized with a seed value generated by the TRNG. The CSE\_SR[EDB] is set if an external debugger is connected. The CSE\_SR[IDB] flag is set when internal debugging has been enabled with the DEBUG\_CHAL and DEBUG\_AUTH commands. The CSE\_SR is a read-only register and can be read at any time.

The Error Code Register (CSE\_ECR) contains the error code from the last completed command. The CSE\_ECR is set to 0 if no error occurred while processing the command. The CSE\_ECR is a read-only register.

## 59.5.2 Command processing

CSE functions are accessed through a set of commands that are listed in [CSE commands](#). Commands are issued by first loading the appropriate parameter registers (CSE\_Px) and then writing a command code to the command register (CSE\_CMD). CSE\_CMD register reads have no effect on the operation of the CSE.

Command inputs and outputs that are 32 bits or less in size are stored directly into a parameter register. Inputs and outputs that are larger than 32 bits are kept in system memory and the address of the data (pointer) is loaded into the parameter register. All data must be aligned on a 32-bit (word) boundary. The two least significant bits of an address are ignored. All data is stored in big-endian format with the most significant byte of the data block stored in the lowest address.

The first command issued to the CSE after reset must be either SECURE\_BOOT or INIT\_CSE. In some device boot modes, system boot logic or BAM code will issue the SECURE\_BOOT command, which initializes the CSE and starts the SHE secure boot protocol (see [Secure boot](#)). For details about how and when the SECURE\_BOOT command is issued on this chip, see the chip-specific CSE information. For the other boot modes, application software must issue the INIT\_CSE command before issuing other CSE commands. The INIT\_CSE command can only be issued one time after reset and only in boot modes that do not issue the SECURE\_BOOT command. Application software cannot issue the SECURE\_BOOT command.

The CSE can only process one command at a time. Commands take a number of clock cycles to complete, so one of the following must be used to determine when the output data is valid and the next command can be issued:

- The CSE\_SR[BSY] flag
- The interrupt request

Parameter registers cannot be modified while the CSE is processing a command. Command processing can be suspended and resumed using the CSE\_CR[SUS] bit (see [Host interface](#)). If the CSE\_CMD register is written while a command is being processed, processing is aborted and an error is generated with CSE\_ECR[EC] = 0x0B. The CANCEL command can be used to abort processing of a command without generating an error. The CSE\_ECR is updated and the CSE\_SR[BSY] bit is cleared upon completion of command processing (see [Error handling](#)).

### 59.5.3 Secure storage

The CSE provides secure, nonvolatile storage for cryptographic keys as described in the *SHE Functional Specification*. The keys are stored in 25 memory slots, with one ROM slot, 23 nonvolatile slots, and one RAM slot as shown in [Table 59-13](#). The first four slots have a dedicated use; the other slots are available for application specific keys. The CSE extends the SHE specification by providing 10 extra application key slots (KEY\_11–KEY\_20) which are enabled by setting the CSE\_CR[KBS] bit. The BOOT\_MAC slot is loaded with a MAC value used by the secure boot process. All other slots are used for encryption or message authentication keys. The SECRET\_KEY slot is programmed with a random value during device fabrication. All CSE encryption and message authentication commands specify a key by its Key ID.

**Table 59-13. Memory slots**

| Slot name      | KBS | Key ID  | Type        |
|----------------|-----|---------|-------------|
| SECRET_KEY     | x   | 0x0     | ROM         |
| MASTER_ECU_KEY | x   | 0x1     | nonvolatile |
| BOOT_MAC_KEY   | x   | 0x2     | nonvolatile |
| BOOT_MAC       | x   | 0x3     | nonvolatile |
| KEY_1–KEY_10   | 0   | 0x4–0xD | nonvolatile |
| KEY_11–KEY_20  | 1   | 0x4–0xD | nonvolatile |
| RAM_KEY        | x   | 0xE     | RAM         |

In addition to the 25 memory slots, the CSE holds a 120-bit read-only unique identification number (UID) that is programmed during device fabrication. The UID is used in the memory update procedure and is available for application-specific uses.

Each memory slot holds a 128-bit value, 6 security flags, and a 28-bit counter. See the *SHE Functional Specification* for the required structure needed for the LOAD\_KEY command. The security flags are defined in [Table 59-14](#). The 28-bit counter must be advanced each time a new key is loaded. The counter value is zero for an empty memory slot.

**Table 59-14. Memory slot security flags**

| Flag name   | Description                                                                                                                                      |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| VERIFY_ONLY | If set, the memory slot key cannot be used by the GENERATE_MAC command, only by the VERIFY_MAC command. This bit has no effect if KEY_USAGE = 0. |
| WRITE_PROT  | If set, the memory slot cannot be updated.                                                                                                       |
| BOOT_PROT   | If set, the memory slot is disabled if CSE_SR[BOK] = 0 or CSE_SR[BFN] = 0.                                                                       |
| DEBUG_PROT  | If set, the memory slot is disabled if CSE_SR[EDB] = 1.                                                                                          |
| KEY_USAGE   | If set, the memory slot holds a MAC key; otherwise, it holds an encryption key.                                                                  |
| WILDCARD    | If set, the memory slot cannot be updated with the wildcard UID.                                                                                 |

The memory slot security flags for a specified key are checked against the associated bits in the CSE\_SR during command processing as described in the SHE specification. If the memory slot is disabled due to security flag settings, a key not available error (EC = 0x03) is returned by the command. The VERIFY\_ONLY security flag is a CSE-specific feature that is not defined in the SHE specification. If the CSE\_CR[SFE] bit is set when the LOAD\_KEY command is executed, the security flag field (FID) in the M2 message is extended by one bit with the VERIFY\_ONLY flag and the adjacent zero fill field is reduced to 94 bits. The CSE\_CR[SFE] bit has no effect on the enforcement of the VERIFY\_ONLY flag for commands issued after the LOAD\_KEY.

A key is loaded into a memory slot using the LOAD\_KEY command which implements the SHE memory update protocol. An empty key slot can only be used as the authorization key to update itself (KeyID = AuthID). In this case, the authorization key has a value of 0. The BOOT\_PROT and DEBUG\_PROT security flags are not enforced for the LOAD\_KEY command except when the RAM\_KEY slot is loaded. In that case, the BOOT\_PROT and DEBUG\_PROT flags of the AuthID slot are applied. Additionally, when loading the RAM\_KEY slot, the counter and security flag fields in the M2 message must be zero.

The LOAD\_PLAIN\_KEY command can be used to load a key into the RAM\_KEY slot. The EXPORT\_RAM\_KEY command can then be used to export the key in an encrypted form compatible with the LOAD\_KEY command for storage outside the CSE. The BOOT\_PROT and DEBUG\_PROT security flags of the MASTER\_ECU\_KEY are enforced for the EXPORT\_RAM\_KEY command.

Two dedicated blocks in the system flash memory are used to implement the secure storage feature. These blocks are not program visible and only the CSE can read, erase, and program these blocks. The CSE reads and writes to the system flash memory via the system bus master interface during the SECURE\_BOOT, INIT\_CSE, DEBUG\_AUTH, and LOAD\_KEY commands. During execution of these commands, the system MPU (if present) must be configured to allow the CSE access to the system flash memory and other bus masters must be programmed properly to avoid interfering with the CSE.

## 59.5.4 Encryption and decryption

The CSE supports AES-128 encryption and decryption in ECB and CBC modes of operation. The key is selected from one of the memory slots, which must be enabled for the operation. A plaintext key can be loaded into the RAM\_KEY slot using the LOAD\_PLAIN\_KEY command for keys that are not stored in a nonvolatile memory slot.



### 59.5.5 Message authentication

The CSE uses the AES-128 CMAC algorithm for message authentication. The key for the CMAC operation is selected from one of the memory slots which must be enabled for the operation. A plaintext key can be loaded into the RAM\_KEY slot using the LOAD\_PLAIN\_KEY command for keys that are not stored in a nonvolatile memory slot. The VERIFY\_MAC command supports comparison of a calculated MAC with an input MAC value.

### 59.5.6 Secure boot

The CSE implements the SHE secure boot protocol. In some device boot modes, the BAM or system boot logic issues the SECURE\_BOOT command to the CSE, which starts the secure boot process. For details about how and when the SECURE\_BOOT command is issued on this chip, see the chip-specific CSE information.

The first step in the process is for the CSE to download the command processor firmware and memory slot data from the CSE flash memory blocks into local memory. If the BOOT\_MAC\_KEY slot is empty, the CSE\_SR[SB] flag is cleared and the process is finished. Otherwise, the CSE\_SR[SB] flag is set and CSE calculates the MAC over the specified bootloader code. If the BOOT\_MAC slot is empty, the calculated MAC is loaded into the BOOT\_MAC slot, the CSE\_SR[BIN] and CSE\_SR[BFN] flags are set, and the process is finished. Otherwise, the calculated MAC value is compared to the value in the BOOT\_MAC slot. If the values match, the CSE\_SR[BOK] bit is set; otherwise, the CSE\_SR[BFN] bit is set. If the CSE\_SR[BOK] flag is set, the user boot code can issue the BOOK\_OK command which sets the CSE\_SR[BFN] bit. The memory slots that have the BOOT\_PROT flag set are enabled when both the CSE\_SR[BFN] and CSE\_SR[BOK] flags are set.

The SECURE\_BOOT command can run either before the user boot code is executed or in parallel with the user boot code. If the parallel mode is used, the user boot code must suspend command processing (set CSE\_CR[SUS] to 1) while performing any operations—such as configuring the flash controller—that may interfere with the CSE accessing flash memory. A device-specific bit selects whether to run the SECURE\_BOOT command before or in parallel with the user boot code. For details about this bit, see the chip-specific CSE information.

The CSE implements an internal timer that is started when the SECURE\_BOOT command is issued. The initial value of the timer is set to 0x00F4\_2400 (16,000,000 CSE system clocks). If the timer expires before the SECURE\_BOOT command completes, any application code intended to be blocked by the SECURE\_BOOT process will instead see the non-completed SECURE\_BOOT status of CSE\_SR[BSY]=1 and

CSE\_SR[BOK]=0. In the event of a SECURE\_BOOT timeout, the CSE will continue executing the SECURE\_BOOT command until completion, until suspended, or until receiving the CANCEL command from the application code. There is no error code associated with a SECURE\_BOOT timeout.

### 59.5.7 Random number generation

The CSE has both a Pseudo Random Number Generator (PRNG) and a True Random Number Generator (TRNG).

The PRNG has a 128-bit state variable and uses AES in output feedback mode to generate pseudo random values. A key derived from the SECRET\_KEY is used for the PRNG. The RND command updates the state of the PRNG and returns the 128-bit random value. The EXTEND\_SEED command can be used to add entropy to the PRNG state. The PRNG state is initialized after each reset with the INIT\_RNG command, which uses the TRNG to generate a 128-bit seed value for the PRNG. The CSE\_SR[RIN] flag is set when the PRNG is initialized.

The INIT\_RNG and TRNG\_RND commands use the TRNG to generate truly random values. The TRNG hardware runs off of a slower clock derived from the system clock. The CSE\_CR[DIV] field needs to be configured for these commands such that the TRNG clock is between 500 kHz and 2 MHz. Random values generated by the TRNG are checked with a statistical test to verify proper operation of the TRNG. If the test fails, a TRNG error (EC = 0x12) is returned. Due to the statistical nature of this test, there is a very small probability ( $< 10^{-9}$ ) that a properly operating TRNG will return an error. If an TRNG error is returned, the command can be issued again.

### 59.5.8 Error handling

When the CSE command processor encounters an error condition, it stops processing and returns an error code as described in [Table 59-15](#). The CSE\_SR[BSY] and CSE\_SR[EX] bits are cleared with interrupt requests generated the same as if the command had completed successfully. In most cases, error conditions are detected before data processing begins and no output values are written. Intermediate or invalid output data is never written to the parameter registers or system memory. However, it is possible for outputs to system memory to be partially written when an error occurs. The CSE does not zero out this partially written data.

**Table 59-15. Error code summary**

| Error code | Error description                  | Error conditions                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|------------|------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x00       | No error                           | 1. Command successfully executed with no errors encountered.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 0x02       | Command sequence error             | <ol style="list-style-type: none"> <li>1. Command issued (except for CANCEL) before the SECURE_BOOT or INIT_CSE command.</li> <li>2. The DEBUG_AUTH command issued before the DEBUG_CHAL command.</li> <li>3. SECURE_BOOT or INIT_CSE command issued after an initial SECURE_BOOT or INIT_CSE is issued.</li> </ol>                                                                                                                                                                                                                                      |
| 0x03       | Key not available                  | 1. A required key is not available due to a BOOT_PROT or DEBUG_PROT security flag restriction (see <a href="#">Secure storage</a> ).                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 0x04       | Invalid key                        | <ol style="list-style-type: none"> <li>1. The specified key slot is not valid for the command. (See the SHE specification.)</li> <li>2. The specified key slot is not available due to a KEY_USAGE security flag restriction (see <a href="#">Secure storage</a>).</li> </ol>                                                                                                                                                                                                                                                                            |
| 0x05       | Empty key                          | 1. The specified key slot is empty.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 0x06       | No secure boot                     | <ol style="list-style-type: none"> <li>1. SECURE_BOOT issued when secure boot is disabled or the BOOT_MAC_KEY slot is empty.</li> <li>2. BOOT_FAIL or BOOT_OK issued with incorrect settings for either the CSE_SR[SB], CSE_SR[BFN] or CSE_SR[BOK] flags.</li> </ol>                                                                                                                                                                                                                                                                                     |
| 0x07       | Key write protected                | <ol style="list-style-type: none"> <li>1. Attempt to load a key slot with the WRITE_PROT security flag set.</li> <li>2. Attempt to enter internal debug mode (DEBUG_CHAL, DEBUG_AUTH) with a WRITE_PROT security flag set in one or more key slots.</li> </ol>                                                                                                                                                                                                                                                                                           |
| 0x08       | Key update error                   | <p>The LOAD_KEY command failed due to one of the following conditions:</p> <ol style="list-style-type: none"> <li>1. The M3 MAC value is invalid.</li> <li>2. The wildcard UID is specified with the WILDCARD flag set.</li> <li>3. The specified UID does not match the device UID.</li> <li>4. The update counter or security flag values are not zero when loading the RAM_KEY slot.</li> <li>5. The specified update counter value is not greater than the current counter value for the slot. (Counter value is zero for an empty slot.)</li> </ol> |
| 0x09       | Random number seed not initialized | 1. RND, EXTEND_SEED, or DEBUG_CHAL command issued before the INIT_RNG command.                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 0x0A       | Internal debug not allowed         | 1. DEBUG_AUTH command issued with invalid MAC value.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 0x0B       | Command issued while busy          | 1. Command issued when the CSE_SR[BSY] bit is set.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 0x0C       | System memory error                | 1. A system memory error was encountered while executing the command. (The CSE flash memory block error code is generated for bus errors encountered when accessing the CSE flash memory blocks.)                                                                                                                                                                                                                                                                                                                                                        |
| 0x10       | Internal memory error              | 1. An internal memory error was encountered while executing the command.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

*Table continues on the next page...*

**Table 59-15. Error code summary (continued)**

| Error code | Error description                | Error conditions                                                                                                                                                       |
|------------|----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0x11       | Invalid command                  | 1. Value written to CSE_CMD register is out of range.                                                                                                                  |
| 0x12       | TRNG error                       | 1. One or more statistical tests run on the TRNG output failed (see <a href="#">Generate Random Number</a> ).                                                          |
| 0x13       | CSE flash memory block error     | 1. Error reading, programming, or erasing one of the CSE flash memory blocks.<br>2. UID or SECRET_KEY required but not available.                                      |
| 0x14       | Internal command processor error | 1. An internal error condition was encountered while executing the command.                                                                                            |
| 0x15       | Length error                     | 1. MAC length for VERIFY_MAC command is greater than 128.<br>2. Message length for GENERATE_MAC, VERIFY_MAC, or MP_COMPRESS command is greater than 0x7FFFFFFF (4 GB). |

## 59.6 CSE commands

This section describes the set of CSE commands. Commands are issued by first loading the appropriate parameter registers (CSE\_Px) and then writing a command code to the command register (CSE\_CMD) as described in [Command processing](#). The first command issued to the CSE after reset must be INIT\_CSE for device boot modes that do not support secure boot. Command inputs and outputs that are 32 bits or less in size are stored directly in a parameter register. Inputs and outputs that are larger than 32 bits are kept in system memory and the address of the data (pointer) is loaded into the parameter register. The data direction indication in the tables below refers to the command parameter value which may be stored in memory.

### 59.6.1 Encrypt ECB

The ENC\_ECB command performs AES-128 encryption in ECB mode on n 128-bit blocks of data with the parameters specified in the table below.

**Table 59-16. ENC\_ECB command**

| Register | Value                | Data direction |
|----------|----------------------|----------------|
| CSE_CMD  | 0x01                 | —              |
| CSE_P1   | Key ID               | Input          |
| CSE_P2   | Number of blocks (n) | Input          |

*Table continues on the next page...*

**Table 59-16. ENC\_ECB command (continued)**

| Register | Value                          | Data direction |
|----------|--------------------------------|----------------|
| CSE_P3   | First plaintext block address  | Input          |
| CSE_P4   | First ciphertext block address | Output         |

## 59.6.2 Encrypt CBC

The ENC\_CBC command performs AES-128 encryption on  $n$  128-bit blocks of data in CBC mode with the parameters specified in the table below. The number of blocks parameter is a 32-bit value.

**Table 59-17. ENC\_CBC command**

| Register | Value                          | Data direction |
|----------|--------------------------------|----------------|
| CSE_CMD  | 0x02                           | —              |
| CSE_P1   | Key ID                         | Input          |
| CSE_P2   | IV address                     | Input          |
| CSE_P3   | Number of blocks ( $n$ )       | Input          |
| CSE_P4   | First plaintext block address  | Input          |
| CSE_P5   | First ciphertext block address | Output         |

## 59.6.3 Decrypt ECB

The DEC\_ECB command performs AES-128 ECB decryption on  $n$  128-bit blocks of data with the parameters specified in the table below.

**Table 59-18. DEC\_ECB command**

| Register | Value                          | Data direction |
|----------|--------------------------------|----------------|
| CSE_CMD  | 0x03                           | —              |
| CSE_P1   | Key ID                         | Input          |
| CSE_P2   | Number of blocks ( $n$ )       | Input          |
| CSE_P3   | First ciphertext block address | Input          |
| CSE_P4   | First plaintext block address  | Output         |

## 59.6.4 Decrypt CBC

The DEC\_CBC command performs AES-128 decryption in CBC mode on  $n$  128-bit blocks of data with the parameters specified in the table below. The number of blocks parameter is a 32-bit value.

**Table 59-19. DEC\_CBC command**

| Register | Value                          | Data direction |
|----------|--------------------------------|----------------|
| CSE_CMD  | 0x04                           | —              |
| CSE_P1   | Key ID                         | Input          |
| CSE_P2   | IV address                     | Input          |
| CSE_P3   | Number of blocks ( $n$ )       | Input          |
| CSE_P4   | First ciphertext block address | Input          |
| CSE_P5   | First plaintext block address  | Output         |

## 59.6.5 Generate MAC

The GENERATE\_MAC command calculates the MAC of a given message with the parameters specified in the table below. The AES CMAC algorithm is used to calculate a 128-bit MAC output. The message length input is a 64-bit value that specifies the length of the message in bits. A length error (EC = 0x15) is returned if the message length is greater than 0x7fffffff (4 GB).

**Table 59-20. GENERATE\_MAC command**

| Register | Value                         | Data direction |
|----------|-------------------------------|----------------|
| CSE_CMD  | 0x05                          | —              |
| CSE_P1   | Key ID                        | Input          |
| CSE_P2   | Message length (bits) address | Input          |
| CSE_P3   | Message start address         | Input          |
| CSE_P4   | MAC address                   | Output         |

## 59.6.6 Verify MAC

The VERIFY\_MAC command verifies the MAC of a given message with the parameters specified in [Table 59-21](#). The AES CMAC algorithm is used to calculate a 128-bit MAC that is truncated according to the MAC length parameter, which specifies the number of most significant bits in the MAC to compare. A MAC length value of 0 indicates that all 128 bits are compared; a value greater than 128 returns a length error (EC = 0x15). The

message length input is a 64-bit value that specifies the length of the message in bits. A length error (EC = 0x15) is returned if the message length is greater than 0x7fffffff (4 GB). If the input MAC matches the MAC calculated over the message, the CSE\_P5 register is set to 0; otherwise, it is set to 1.

**Table 59-21. VERIFY\_MAC command**

| Register | Value                                     | Data direction    |
|----------|-------------------------------------------|-------------------|
| CSE_CMD  | 0x06                                      | —                 |
| CSE_P1   | Key ID                                    | Input             |
| CSE_P2   | Message length (bits) address             | Input             |
| CSE_P3   | Message start address                     | Input             |
| CSE_P4   | MAC address                               | Input             |
| CSE_P5   | MAC length (bits)/<br>Verification status | Input /<br>Output |

### 59.6.7 Load Key

The LOAD\_KEY command updates a memory slot using parameters specified in [Table 59-22](#) according to the SHE memory slot update protocol (see [Secure storage](#)). The 128-bit M1 message contains the UID, Key ID, and Authentication Key ID. The 256-bit M2 message contains the new security flags, the counter, and the key value all encrypted using a derived key generated from the Authentication Key. The 128-bit M3 message is a MAC generated over messages M1 and M2. The 256-bit M4 message is the concatenation of the UID, Key ID, Authorization Key ID, and the encrypted counter value. The 128-bit M5 message is the MAC calculated over message M4.

The derived keys used to generate messages M2, M3, M4, and M5 deviate from the SHE memory slot update protocol when updating the second bank of user keys (KEY\_11–KEY\_20). During the generation of M2 and M4, K1 and K3 shall be derived using a constant equal to KEY\_UPDATE\_ENC\_C + 0x00800000\_00000000\_00000000\_00000000. Similarly, during generation of M3 and M5, K2 and K4 shall be derived using a constant equal to KEY\_UPDATE\_MAC\_C + 0x00800000\_00000000\_00000000\_00000000. This is intended to prevent aliasing between the two banks of user keys.

**Table 59-22. LOAD\_KEY command**

| Register | Value      | Data direction |
|----------|------------|----------------|
| CSE_CMD  | 0x07       | —              |
| CSE_P1   | M1 address | Input          |

*Table continues on the next page...*

**Table 59-22. LOAD\_KEY command (continued)**

| Register | Value      | Data direction |
|----------|------------|----------------|
| CSE_P2   | M2 address | Input          |
| CSE_P3   | M3 address | Input          |
| CSE_P4   | M4 address | Output         |
| CSE_P5   | M5 address | Output         |

### 59.6.8 Load Plain Key

The LOAD\_PLAIN\_KEY command updates the RAM key memory slot with a 128-bit plaintext key with the parameter shown in the table below.

**Table 59-23. LOAD\_PLAIN\_KEY command**

| Register | Value       | Data direction |
|----------|-------------|----------------|
| CSE_CMD  | 0x08        | —              |
| CSE_P1   | Key address | Input          |

### 59.6.9 Export RAM Key

The EXPORT\_RAM\_KEY command exports the RAM key data with the parameters in the table below. Only keys loaded with the LOAD\_PLAIN\_KEY command may be exported. The output messages are compatible with the messages used for LOAD\_KEY (see [Load Key](#) for details).

**Table 59-24. EXPORT\_RAM\_KEY command**

| Register | Value      | Data direction |
|----------|------------|----------------|
| CSE_CMD  | 0x09       | —              |
| CSE_P1   | M1 address | Output         |
| CSE_P2   | M2 address | Output         |
| CSE_P3   | M3 address | Output         |
| CSE_P4   | M4 address | Output         |
| CSE_P5   | M5 address | Output         |



## 59.6.10 Initialize RNG

The INIT\_RNG command initializes the internal PRNG state with a seed value generated by the TRNG and sets the CSE\_SR[RIN] flag. It takes 2048 TRNG clock cycles to generate a seed value. The CSE\_CR[DIV] field must be properly configured before this command is executed (see register descriptions for further information). There is a very small probability that this command may return a TRNG error (EC = 0x12) even when the TRNG is operating properly (see [Random number generation](#)). The INIT\_RNG command must be called after each reset before the RND command is issued. The command has no parameters as shown the table below.

**Table 59-25. INIT\_RNG command**

| Register | Value | Data direction |
|----------|-------|----------------|
| CSE_CMD  | 0x0A  | —              |

## 59.6.11 Extend PRNG Seed

The EXTEND\_SEED command extends the state of the PRNG using a 128-bit entropy input value. The current PRNG state and the input data are compressed into a new PRNG state value. This command is issued with the parameter shown in the table below.

**Table 59-26. EXTEND\_SEED command**

| Register | Value                 | Data direction |
|----------|-----------------------|----------------|
| CSE_CMD  | 0x0B                  | —              |
| CSE_P1   | Entropy value address | Input          |

## 59.6.12 Generate Random Number

The RND command generates a 128-bit random value and updates the state of the internal PRNG. The PRNG state must be initialized after reset using the INIT\_RNG command before this command can be issued. This command is issued with the parameter shown in the table below

**Table 59-27. RND command**

| Register | Value                | Data direction |
|----------|----------------------|----------------|
| CSE_CMD  | 0x0C                 | —              |
| CSE_P1   | Random value address | Output         |

### 59.6.13 Secure Boot

The SECURE\_BOOT command loads the command processor firmware and memory slot data from the CSE flash memory blocks, and then it executes the SHE secure boot protocol using the parameters shown in [Table 59-28](#) (see [Secure boot](#) for details). The bootloader size is a 32-bit value. The CSE firmware version is loaded into CSE\_P3 register.

The SECURE\_BOOT command is issued during the boot processes by the system boot logic. For details about how and when the SECURE\_BOOT command is issued on this chip, see the chip-specific CSE information.

**Table 59-28. SECURE\_BOOT command**

| Register | Value                    | Data direction |
|----------|--------------------------|----------------|
| CSE_CMD  | 0x0D                     | —              |
| CSE_P1   | Bootloader size (bytes)  | Input          |
| CSE_P2   | Bootloader start address | Input          |
| CSE_P3   | Firmware version         | Output         |

### 59.6.14 Boot Failure

The BOOT\_FAILURE command sets the CSE\_SR[BFN] flag and clears the CSE\_SR[BOK] flag to disable use of memory slots that have the BOOT\_PROT flag set. This command is issued with no parameters as shown in the table below.

**Table 59-29. BOOT\_FAILURE command**

| Register | Value | Data direction |
|----------|-------|----------------|
| CSE_CMD  | 0x0E  | —              |

### 59.6.15 Boot OK

The BOOT\_OK command sets the CSE\_SR[BFN] flag and leaves the CSE\_SR[BOK] flag set to confirm successful completion of the secure boot processes. This enables the use of memory slots that have the BOOT\_PROT flag set if they are not disabled for some other reason. This command is issued with no parameters as shown in the table below.

**Table 59-30. BOOT\_OK command**

| Register | Value | Data direction |
|----------|-------|----------------|
| CSE_CMD  | 0x0F  | —              |

### 59.6.16 Get ID

The GET\_ID command returns the UID, CSE\_SR[24:31], and a 128-bit MAC calculated over the concatenation of a 128-bit input challenge value, UID, and CSE\_SR[24:31]. The MASTER\_ECU\_KEY is used for the MAC calculation. A value of 0 is returned for the MAC if the MASTER\_ECU\_KEY slot is empty. The UID output is a 128-bit value with the 8 least significant bits set to 0. This command is issued with the parameters shown in the table below.

**Table 59-31. GET\_ID command**

| Register | Value               | Data direction |
|----------|---------------------|----------------|
| CSE_CMD  | 0x10                | —              |
| CSE_P1   | Challenge address   | Input          |
| CSE_P2   | UID output address  | Output         |
| CSE_P3   | CSE_SR[24:31] value | Output         |
| CSE_P4   | MAC address         | Output         |

### 59.6.17 Cancel

The CANCEL command aborts processing of the current command and clears the CSE\_SR[BSY] flag. This command is issued with no parameters as shown in the table below.

**Table 59-32. CANCEL command**

| Register | Value | Data direction |
|----------|-------|----------------|
| CSE_CMD  | 0x11  | —              |

## 59.6.18 Debug Challenge

The `DEBUG_CHAL` command generates a 128-bit random challenge output value that is used in conjunction with the `DEBUG_AUTH` command. The PRNG state must be initialized after reset using the `INIT_RNG` command before this command can be issued. This command is issued with the parameter shown in the table below.

**Table 59-33. `DEBUG_CHAL` command**

| Register | Value             | Data direction |
|----------|-------------------|----------------|
| CSE_CMD  | 0x12              | —              |
| CSE_P1   | Challenge address | Output         |

## 59.6.19 Debug Authorization

The `DEBUG_AUTH` command erases all user keys and sets the `CSE_SR[IDB]` flag which enables internal debugging if the 128-bit authorization input value is valid and no memory slots are write protected. The authorization input is generated using the `DEBUG_CHAL` command output and the UID as described in the *SHE Functional Specification*. If the `DEBUG_CHAL` command is not issued before the `DEBUG_AUTH` command, a command sequence error (`EC = 0x02`) is returned. However, other commands may be issued between the `DEBUG_CHAL` and `DEBUG_AUTH` commands. This command is issued with the parameter shown in the table below.

**Table 59-34. `DEBUG_AUTH` command**

| Register | Value                 | Data direction |
|----------|-----------------------|----------------|
| CSE_CMD  | 0x13                  | —              |
| CSE_P1   | Authorization address | Input          |

## 59.6.20 Generate TRNG Random Number

The `TRNG_RND` command generates a 128-bit random output value using the TRNG. It takes 2048 TRNG clock cycles to generate a random value. The `CSE_CR[DIV]` field must be properly configured before this command is executed (see the register descriptions section for further information). This command takes much longer to execute than the `RND` command that should normally be used to generate random values. There is a very small probability that this command may return a TRNG error (`EC = 0x12`) even when the TRNG is operating properly (see [Random number generation](#)). This command is issued with the parameter shown in the table below.

**Table 59-35. TRNG\_RND Command**

| Register | Value                | Data direction |
|----------|----------------------|----------------|
| CSE_CMD  | 0x14                 | —              |
| CSE_P1   | Random value address | Output         |

### 59.6.21 Initialize CSE

The INIT\_CSE command loads the command processor firmware and memory slot data from the CSE flash memory blocks into local memory. It does not execute the secure boot protocol. The CSE firmware version is loaded into the CSE\_P1 register as shown in the table below. This command must be issued before any other command when secure boot is not enabled.

**Table 59-36. INIT\_CSE Command**

| Register | Value            | Data direction |
|----------|------------------|----------------|
| CSE_CMD  | 0x15             | —              |
| CSE_P1   | Firmware version | Output         |

### 59.6.22 Miyaguchi-Preneel (MP) Compression

The MP\_COMPRESS command is a one-way compression function used to derive a 128-bit output from a given message with the parameters specified in [Table 59-37](#). The AES ECB algorithm is used to derive the 128-bit output as described in the SHE specification. The message length input is a 64-bit value which specifies the length of the message in bits. A length error (EC=0x15) is returned if the message length is greater than 0x7fffffff (4GB).

**Table 59-37. MP\_COMPRESS Command**

| Register | Value                         | Data direction |
|----------|-------------------------------|----------------|
| CSE_CMD  | 0x016                         |                |
| CSE_P1   | Message length (bits) address | Input          |
| CSE_P2   | Message start address         | Input          |
| CSE_P3   | Output address                | Output         |



# Chapter 60

## Password and Device Security Module (PASS)

### 60.1 Chip-specific PASS information

This chip has one instance of the PASS module. PASS implements password-based read and write protection for flash memory blocks.

This chip implements four protection regions.

#### 60.1.1 Accesses to unimplemented PASS registers

On this chip, accesses to unimplemented PASS registers generate bus aborts.

#### 60.1.2 PASS\_LOCKm\_PGn register bit mapping

Each password group defined in the DCF records is associated with a set of four PASS\_LOCKm\_PGn registers. Each implemented bit in these registers maps to a specific memory block.

**NOTE**

These registers are initialized via DCF records, so the reset values depend on user settings.

##### 60.1.2.1 PASS\_LOCK0\_PGn register bit mapping

The flash memory blocks mapped to the [Password Group n - Lock 0 Status Register \(PASS\\_LOCK0\\_PGn\)](#) bits, and the corresponding PASS protection regions, are as follows.

|   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|

Table continues on the next page...

| Protection Region                  | Flash Memory Block Name                | Register Field |          |
|------------------------------------|----------------------------------------|----------------|----------|
|                                    |                                        | TSLOCK         | Reserved |
| PASS Read Lock Protection Region 0 | UTest NVM Block Space 16 KB            | LOWLOCK[13:0]  |          |
|                                    | —                                      |                |          |
|                                    | —                                      |                |          |
|                                    | —                                      |                |          |
|                                    | —                                      |                |          |
|                                    | —                                      |                |          |
|                                    | —                                      |                |          |
|                                    | —                                      |                |          |
|                                    | —                                      |                |          |
|                                    | —                                      |                |          |
|                                    | —                                      |                |          |
|                                    | —                                      |                |          |
|                                    | —                                      |                |          |
|                                    | —                                      |                |          |
| PASS Read Lock Protection Region 2 | 64 KB EEPROM Data Block1 - Partition 1 | MIDLOCK[15:0]  |          |
| PASS Read Lock Protection Region 2 | 64 KB EEPROM Data Block0 - Partition 0 |                |          |
|                                    | —                                      |                |          |
|                                    | —                                      |                |          |
|                                    | —                                      |                |          |
|                                    | —                                      |                |          |
|                                    | —                                      |                |          |
|                                    | —                                      |                |          |
|                                    | —                                      |                |          |
|                                    | —                                      |                |          |
|                                    | —                                      |                |          |
|                                    | —                                      |                |          |
|                                    | —                                      |                |          |
|                                    | —                                      |                |          |
| PASS Read Lock Protection Region 2 | 64 KB EEPROM Data Block1 - Partition 3 | MIDLOCK[15:0]  |          |
| PASS Read Lock Protection Region 2 | 64 KB EEPROM Data Block0 - Partition 2 |                |          |
|                                    | —                                      |                |          |
|                                    | —                                      |                |          |
|                                    | —                                      |                |          |
|                                    | —                                      |                |          |
|                                    | —                                      |                |          |
|                                    | —                                      |                |          |
|                                    | —                                      |                |          |
|                                    | —                                      |                |          |
|                                    | —                                      |                |          |
|                                    | —                                      |                |          |
|                                    | —                                      |                |          |
|                                    | —                                      |                |          |
|                                    | —                                      |                |          |

### 60.1.2.2 PASS\_LOCK1\_PG $n$ register bit mapping

The flash memory blocks mapped to the [Password Group  \$n\$  - Lock 1 Status Register \(PASS\\_LOCK1\\_PG \$n\$ \)](#) bits, and the corresponding PASS protection region, are as follows.

#### NOTE

This register cannot be used to manage the blocks mapped to the register's implemented bits because the CSE module has exclusive access to these blocks. The bits' value is 1 to reflect



the corresponding blocks' locked state, and the bits cannot be written to change that state.

| 012345678910111213141516171819202122232425262728293031 |                                     |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|--------------------------------------------------------|-------------------------------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|----------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| Register Field                                         | Reserved                            |  |  |  |  |  |  |  |  |  |  |  |  |  |  | HIGHLOCK[15:0] |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Protection Region                                      | Flash Memory Block Name             |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| —                                                      | —                                   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| —                                                      | —                                   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| —                                                      | —                                   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| —                                                      | —                                   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| —                                                      | —                                   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| —                                                      | —                                   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| —                                                      | —                                   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| —                                                      | —                                   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| —                                                      | —                                   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| —                                                      | —                                   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| —                                                      | —                                   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| —                                                      | —                                   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| —                                                      | —                                   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| —                                                      | —                                   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| —                                                      | —                                   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| —                                                      | —                                   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| —                                                      | —                                   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| —                                                      | —                                   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| —                                                      | —                                   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| —                                                      | —                                   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| —                                                      | —                                   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| —                                                      | —                                   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| —                                                      | —                                   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| —                                                      | —                                   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| —                                                      | —                                   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| —                                                      | —                                   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| —                                                      | —                                   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| —                                                      | —                                   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| —                                                      | —                                   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| PASS Read Lock Protection Region 3                     | 16 KB CSE Data Block1 - Partition 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| PASS Read Lock Protection Region 3                     | 16 KB CSE Data Block0 - Partition 4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

### 60.1.2.3 PASS\_LOCK2\_PGn register bit mapping

The flash memory blocks mapped to the Password Group  $n$  - Lock 2 Status Register (PASS\_LOCK2\_PG $n$ ) bits, and the corresponding PASS protection region, are as follows.

\_\_\_\_\_

[illegible]

#### 60.1.2.4 PASS\_LOCK3\_PGn register bit mapping

The flash memory blocks mapped to the Password Group  $n$  - Lock 3 Status Register (PASS\_LOCK3\_PG $n$ ) bits, and the corresponding PASS protection region, are as follows.

Reserved bits read as 0.

|   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|

*Table continues on the next page...*

[illegible]

### 60.1.3 PASS chip select (client select)

On this chip, the chip select (client select) for the PASS module is CS[3].

For details about CS[3], see the DCF Records chapter.

#### 60.1.4 Flash memory read protection scenarios

The following table summarizes the scenarios of read lock (RL) region protection on this chip. For more information, see [Flash Memory Read Protection](#).

### Table 60-1. Flash memory read protection truth table

| Inputs                     |                       |                 |            | Outputs                      |                                                        |
|----------------------------|-----------------------|-----------------|------------|------------------------------|--------------------------------------------------------|
| Life Cycle                 | Censorship DCF client | Debug Interface | LOCK3[RLx] | Read Protected RL[0] (UTest) | Read Protected RL[3:1] (other regions) <sup>1, 2</sup> |
| Customer Delivery          | Don't care            | Don't care      | Don't care | Readable                     | Readable                                               |
| OEM Production or In Field | 0x55AA                | Don't care      | Don't care | Readable                     | Readable                                               |
|                            | Any other value       | Blocked         | Don't care | Readable                     | Readable                                               |
|                            |                       | Enabled         | Locked     | Read Blocked <sup>34</sup>   | Read Blocked <sup>3, 4</sup>                           |

*Table continues on the next page...*

**Table 60-1. Flash memory read protection truth table (continued)**

| Inputs           |                       |                 |            | Outputs                      |                                                        |
|------------------|-----------------------|-----------------|------------|------------------------------|--------------------------------------------------------|
| Life Cycle       | Censorship DCF client | Debug Interface | LOCK3[RLx] | Read Protected RL[0] (UTest) | Read Protected RL[3:1] (other regions) <sup>1, 2</sup> |
|                  |                       |                 | Unlocked   | Readable                     | Readable                                               |
| Failure Analysis | 0x55AA                | Don't care      | Don't care | Readable                     | Readable                                               |
|                  | Any other value       | Don't care      | Locked     | Readable                     | Read Blocked                                           |
|                  |                       |                 | Unlocked   | Readable                     | Readable                                               |

1. When Life Cycle is other than In Field: If the BAM cannot find a valid RCHW because the corresponding regions are Read Blocked, the boot mode is switched to serial.
2. After the Life Cycle state OEM Production, the UTest space reserved for passwords is protected by the flash memory interface instead of PASS. PASS has no effect on the space.
3. Flash memory is readable if the JTAG password is used to break the censorship scheme.
4. When Life Cycle is In Field, the BAM will not switch to serial boot mode. Instead, it will keep searching for a valid RCHW until SWT requests a reset.

## 60.1.5 Debug Interface Access

Whether the Debug Interface Access is enabled or blocked depends on multiple input conditions. The following table shows the the Debug Interface Access scenarios on this chip.

**Table 60-2. Debug Interface Access truth table**

| Life Cycle                    | Production Disable | Censorship DCF client | Debug Enable LOCK3[DBL] bit | JTAG Password | Debug Interface Access status |
|-------------------------------|--------------------|-----------------------|-----------------------------|---------------|-------------------------------|
| Customer Delivery             | Don't care         | Don't care            | Don't care                  | Don't care    | Enabled                       |
| OEM Production                | Don't care         | 0x55AA                | Don't care                  | Don't care    | Enabled                       |
|                               |                    | Any other value       | Unlocked (DBL = 0)          | Don't care    | Enabled                       |
|                               |                    |                       | Locked (DBL = 1)            | Matched       | Enabled                       |
|                               |                    |                       |                             | Not Matched   | Blocked                       |
| In Field                      | Off                | 0x55AA                | Don't care                  | Don't care    | Enabled                       |
|                               |                    | Any other value       | Unlocked (DBL = 0)          | Don't care    | Enabled                       |
|                               |                    |                       | Locked (DBL = 1)            | Matched       | Enabled                       |
|                               |                    |                       |                             | Not Matched   | Blocked                       |
|                               |                    |                       |                             | Not Matched   | Blocked                       |
|                               | On                 | Don't care            | Don't care                  | Don't care    | Blocked                       |
| Failure Analysis <sup>1</sup> | Don't care         | 0x55AA                | Don't care                  | Don't care    | Enabled                       |
|                               |                    | Any other value       | Unlocked (DBL = 0)          | Don't care    | Enabled                       |
|                               |                    |                       | Locked (DBL = 1)            | Matched       | Enabled                       |
|                               |                    |                       |                             | Not Matched   | Blocked                       |

1. When advancing the Life Cycle to FA, program the Censorship DCF client with 0x55AA to enable the Debug Interface Access for Freescale.

When the Life Cycle is Customer Delivery, the Debug Interface Access is fully enabled regardless of any boundary conditions.

When the Life Cycle is OEM Production, In Field, or Failure Analysis, the Debug Interface Access depends on the Censorship DCF client settings, the 256-bit JTAG Password entered via JTAG mechanism, and the LOCK3[DBL] bit.

For the Life Cycle of In Field, the Debug Interface is further conditioned by the Production Disable feature. If the Production Disable feature is enabled when the Life Cycle is In Field, the Debug Interface Access may be permanently blocked because Production Disable overrides all other debug controls. The preceding table emphasizes this scenario by shading the set of conditions and the result in grey.

### 60.1.6 Duration of Debug Interface Access via JTAG password

[JTAG Password Challenge](#) explains the method for enabling Debug Interface Access by writing the [JTAG\\_PASSWORD register](#) via an external tool. Debug Interface Access granted by a password match remains enabled until the next "destructive" reset. On this chip, a "destructive" reset is any of the following:

- POR
- A reset caused by:
  - LVD or HVD event
  - Temperature threshold detection event
  - RESET pin
  - STCU2 self test
  - FOSU timeout

### 60.1.7 Production Disable enablement

The enablement of the Production Disable feature is done via a DCF client that is not accessible by software and that is automatically loaded during the reset phase of the device. For more information, see [Production disable](#).

#### CAUTION

Be careful when configuring the Production Disable feature because enabling Production Disable might permanently block the Debug Interface Access.

### 60.1.8 PASS\_LCSTAT[LIFE] value source

On this chip, the SSCM is the "external" module from which PASS\_LCSTAT[LIFE] derives its reset value. The value of PASS\_LCSTAT[LIFE] directly reflects the value of SSCM\_LCSTAT[LC].

## 60.2 Introduction

### 60.2.1 Overview

The Password and device security (PASS) module shown in the following figure, receives password challenges and determines their validity. It also maintains the device security and access states.

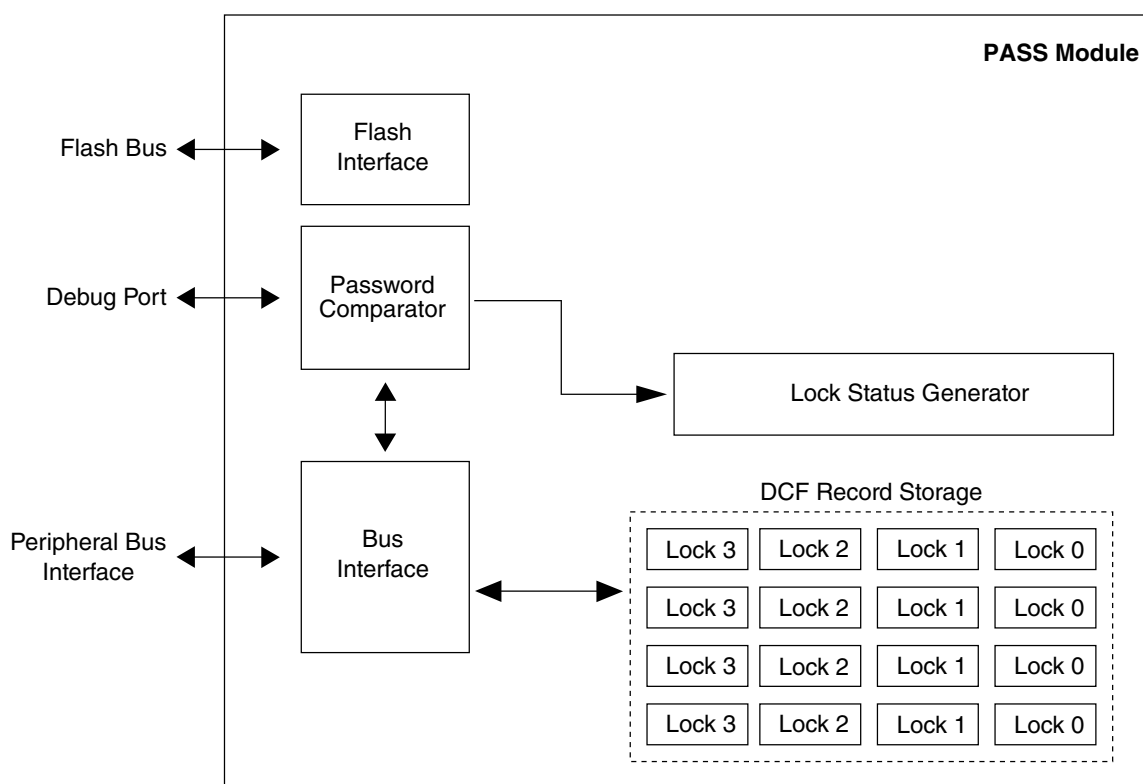


Figure 60-1. PASS module block diagram

### 60.2.2 Features

The PASS includes these distinctive features:

- Life Cycle status
- Password comparison
- JTAG password comparison
- Production Disable control
- Flash read, write, and erase control

### 60.2.3 Modes of operation

The PASS operates identically in all system modes.

## 60.3 Memory map and register definition

This section provides a detailed description of all memory-mapped registers and DCF clients in the PASS.

Where a DCF client is listed, the reset value of that register may automatically be pre-loaded during reset based on the data stored in DCF Records in flash UTEST region.

The following table shows the memory map for the PASS. Note that all addresses are offsets; the absolute address for registers may be calculated by adding the specified offset to the base address of the PASS. The address of the DCF client is local to the PASS module and must be qualified with the appropriate Client Select (CS[14:0]) for the PASS module, as defined in the DCF Record Section of the Reference Manual.

#### NOTE

Non-implemented registers generate bus aborts if enabled.

The following registers are available in the PASS. These bit fields are chip-specific and user should refer to PASS chip-specific chapter to see the flash memory regions to which they apply the shaded bits are reserved for future use. For future compatibility, only write '0' to these bits; they always read '0'.

**PASS memory map**

| Address offset (hex) | Register name                            | Width (in bits) | Access | Reset value                 | Section/ page               |
|----------------------|------------------------------------------|-----------------|--------|-----------------------------|-----------------------------|
| 0                    | Life Cycle Status Register (PASS_LCSTAT) | 32              | R      | <a href="#">See section</a> | <a href="#">60.3.1/2725</a> |
| 8                    | Challenge Selector Register (PASS_CHSEL) | 32              | R/W    | 0000_0000h                  | <a href="#">60.3.2/2726</a> |
| 10                   | Challenge Status Register (PASS_CSTAT)   | 32              | R      | 0000_0000h                  | <a href="#">60.3.3/2727</a> |

*Table continues on the next page...*

**PASS memory map (continued)**

| Address offset (hex) | Register name                                              | Width (in bits) | Access                | Reset value                 | Section/ page               |
|----------------------|------------------------------------------------------------|-----------------|-----------------------|-----------------------------|-----------------------------|
| 20                   | Challenge Input Register (PASS_CIN0)                       | 32              | W<br>(always reads 0) | 0000_0000h                  | <a href="#">60.3.4/2727</a> |
| 24                   | Challenge Input Register (PASS_CIN1)                       | 32              | W<br>(always reads 0) | 0000_0000h                  | <a href="#">60.3.4/2727</a> |
| 28                   | Challenge Input Register (PASS_CIN2)                       | 32              | W<br>(always reads 0) | 0000_0000h                  | <a href="#">60.3.4/2727</a> |
| 2C                   | Challenge Input Register (PASS_CIN3)                       | 32              | W<br>(always reads 0) | 0000_0000h                  | <a href="#">60.3.4/2727</a> |
| 30                   | Challenge Input Register (PASS_CIN4)                       | 32              | W<br>(always reads 0) | 0000_0000h                  | <a href="#">60.3.4/2727</a> |
| 34                   | Challenge Input Register (PASS_CIN5)                       | 32              | W<br>(always reads 0) | 0000_0000h                  | <a href="#">60.3.4/2727</a> |
| 38                   | Challenge Input Register (PASS_CIN6)                       | 32              | W<br>(always reads 0) | 0000_0000h                  | <a href="#">60.3.4/2727</a> |
| 3C                   | Challenge Input Register (PASS_CIN7)                       | 32              | W<br>(always reads 0) | 0000_0000h                  | <a href="#">60.3.4/2727</a> |
| 100                  | Password Group n - Lock 0 Status Register (PASS_LOCK0_PG0) | 32              | R/W                   | <a href="#">See section</a> | <a href="#">60.3.5/2728</a> |
| 104                  | Password Group n - Lock 1 Status Register (PASS_LOCK1_PG0) | 32              | R/W                   | <a href="#">See section</a> | <a href="#">60.3.6/2730</a> |
| 108                  | Password Group n - Lock 2 Status Register (PASS_LOCK2_PG0) | 32              | R/W                   | <a href="#">See section</a> | <a href="#">60.3.7/2731</a> |
| 10C                  | Password Group n - Lock 3 Status Register (PASS_LOCK3_PG0) | 32              | R/W                   | <a href="#">See section</a> | <a href="#">60.3.8/2732</a> |
| 110                  | Password Group n - Lock 0 Status Register (PASS_LOCK0_PG1) | 32              | R/W                   | <a href="#">See section</a> | <a href="#">60.3.5/2728</a> |
| 114                  | Password Group n - Lock 1 Status Register (PASS_LOCK1_PG1) | 32              | R/W                   | <a href="#">See section</a> | <a href="#">60.3.6/2730</a> |
| 118                  | Password Group n - Lock 2 Status Register (PASS_LOCK2_PG1) | 32              | R/W                   | <a href="#">See section</a> | <a href="#">60.3.7/2731</a> |
| 11C                  | Password Group n - Lock 3 Status Register (PASS_LOCK3_PG1) | 32              | R/W                   | <a href="#">See section</a> | <a href="#">60.3.8/2732</a> |
| 120                  | Password Group n - Lock 0 Status Register (PASS_LOCK0_PG2) | 32              | R/W                   | <a href="#">See section</a> | <a href="#">60.3.5/2728</a> |
| 124                  | Password Group n - Lock 1 Status Register (PASS_LOCK1_PG2) | 32              | R/W                   | <a href="#">See section</a> | <a href="#">60.3.6/2730</a> |
| 128                  | Password Group n - Lock 2 Status Register (PASS_LOCK2_PG2) | 32              | R/W                   | <a href="#">See section</a> | <a href="#">60.3.7/2731</a> |

Table continues on the next page...



**PASS memory map (continued)**

| Address offset (hex) | Register name                                              | Width (in bits) | Access | Reset value                 | Section/ page               |
|----------------------|------------------------------------------------------------|-----------------|--------|-----------------------------|-----------------------------|
| 12C                  | Password Group n - Lock 3 Status Register (PASS_LOCK3_PG2) | 32              | R/W    | <a href="#">See section</a> | <a href="#">60.3.8/2732</a> |
| 130                  | Password Group n - Lock 0 Status Register (PASS_LOCK0_PG3) | 32              | R/W    | <a href="#">See section</a> | <a href="#">60.3.5/2728</a> |
| 134                  | Password Group n - Lock 1 Status Register (PASS_LOCK1_PG3) | 32              | R/W    | <a href="#">See section</a> | <a href="#">60.3.6/2730</a> |
| 138                  | Password Group n - Lock 2 Status Register (PASS_LOCK2_PG3) | 32              | R/W    | <a href="#">See section</a> | <a href="#">60.3.7/2731</a> |
| 13C                  | Password Group n - Lock 3 Status Register (PASS_LOCK3_PG3) | 32              | R/W    | <a href="#">See section</a> | <a href="#">60.3.8/2732</a> |

**60.3.1 Life Cycle Status Register (PASS\_LCSTAT)**

The Life Cycle Status Register reflects the production status and DCF Censorship client status of the device. It is possible to mature the device (move the device life cycle forward), but never to revert the life cycle to an earlier state.

The reset value of the LCSTAT register LIFE field derives from an external life cycle management module (like SSCM). It is a 3-bit life cycle. This register simply displays the life cycle input from external blocks.

Address: 0h base + 0h offset = 0h

|       |     |     |    |    |    |    |    |    |    |    |    |    |    |      |    |    |
|-------|-----|-----|----|----|----|----|----|----|----|----|----|----|----|------|----|----|
| Bit   | 0   | 1   | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13   | 14 | 15 |
| R     | CNS | JUN | 1  | 0  |    |    |    |    |    |    |    |    |    |      |    |    |
| W     |     |     |    |    |    |    |    |    |    |    |    |    |    |      |    |    |
| Reset | *   | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0  | 0  |
| Bit   | 16  | 17  | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29   | 30 | 31 |
| R     | 0   |     |    |    |    |    |    |    |    |    |    |    |    | LIFE |    |    |
| W     |     |     |    |    |    |    |    |    |    |    |    |    |    |      |    |    |
| Reset | 0   | 0   | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | *    | *  | *  |

\* Notes:

- LIFE field: Reset value is the Life Cycle of the device, and is unaffected by reset.
- CNS field: Reset value depends on the Censorship DCF client

**PASS\_LCSTAT field descriptions**

| Field    | Description                                                                                                                                                                                                                                 |
|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>CNS | This field indicates whether the DCF Censorship client has been written with a DCF record to 0x55AA or if the DCF Censorship client contains any other value than 0x55AA.<br>The bit can be set (or cleared) via DCF Record in UTest Flash. |

*Table continues on the next page...*

**PASS\_LCSTAT field descriptions (continued)**

| Field            | Description                                                                                                                                                                                                                                                  |
|------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                  | <b>NOTE:</b> If no DCF record has been written to affect the value of the CNS bit, its reset value will be 1.<br>1 DCF Censorship Client value is anything other than 0x55AA.<br>0 DCF Censorship client value is 0x55AA.                                    |
| 1<br>JUN         | This field indicates whether the device has been temporarily uncensored. Successful transmission of the JTAG password sets this bit to '1', otherwise it will be 0.<br>Unsuccessful transmission clears this bit.                                            |
| 2<br>Reserved    | This field is reserved.<br>This read-only field is reserved and always has the value 1.                                                                                                                                                                      |
| 3–28<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                      |
| 29–31<br>LIFE    | The life cycle setting is manipulated externally (SSCM). Pass displays a 3-bit input life cycle<br>000 Failure Analysis<br>001 Reserved<br>010 OEM Production<br>011 Customer Delivery<br>100 Reserved<br>101 Reserved<br>110 MCU Production<br>111 In Field |

**60.3.2 Challenge Selector Register (PASS\_CHSEL)**

This register determines which password group is challenged via the Challenge Input Register.

When the Master Only (MO) bit of the PASS\_LOCK3\_PGn register is set, only the master that unlocked access to the passgroup settings will be able to access these registers. The CMST field of the Challenge Status Register will reflect the locking master. When the Master Only (MO) bit of the PASS\_LOCK3\_PGn register is not set, the passgroup settings can be accessed by any master.

When a master writes this register, its ID sets the CMST field of the Challenge Status register.

Address: 0h base + 8h offset = 8h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**PASS\_CHSEL field descriptions**

| Field            | Description                                                                                                              |
|------------------|--------------------------------------------------------------------------------------------------------------------------|
| 0–29<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                  |
| 30–31<br>GRP     | Password Group to unlock<br><br>00 Password Group 0<br>01 Password Group 1<br>10 Password Group 2<br>11 Password Group 3 |

**60.3.3 Challenge Status Register (PASS\_CSTAT)**

This register provides status information for the password challenge.

Only the master shown in this register can access the CINn registers - access by other masters result in a comparison fail regardless of the values written.

Address: 0h base + 10h offset = 10h

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27   | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | CMST |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0    | 0  | 0  | 0  |    |

**PASS\_CSTAT field descriptions**

| Field            | Description                                                                             |
|------------------|-----------------------------------------------------------------------------------------|
| 0–27<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0. |
| 28–31<br>CMST    | ID of the master which has last written the CHSEL register                              |

**60.3.4 Challenge Input Register (PASS\_CINn)**

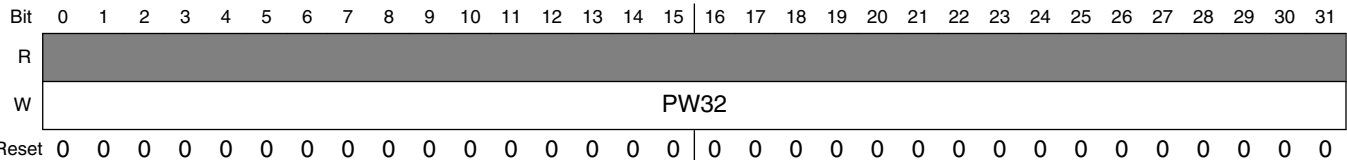
This register is used to provide the Challenge word *n* for a password group as selected by the Challenge Selector Register.

After programming the Challenge Selector Register, the password challenge needs to be written with eight 32-bit writes from CIN0 to CIN7 in that order. The most significant 32 bits of the password challenge must be written to CIN0. Once CIN7 is written, the password is compared. The write access only completes after the comparison has been

executed. In case of a successful password comparison, the PASS module stores the ID of the master which has written CIN0 to CIN7 in the PASS\_LOCK3\_PGn register of the selected password group, as well as clears the PGL bit of that register.

All accesses to these registers need to be made by the bus master that wrote the CHSEL register previously. If another master writes any of the CIN registers before the original master has written CIN7, the comparison fails.

Address: 0h base + 20h offset + (4d × i), where i=0d to 7d



PASS\_CINn field descriptions

| Field        | Description                                                                |
|--------------|----------------------------------------------------------------------------|
| 0–31<br>PW32 | 32 bits of the 256-bit password challenge. Only 32-bit writes may be used. |

60.3.5 Password Group n - Lock 0 Status Register (PASS\_LOCK0\_PGn)

Each Password Group has one Lock 0 Status register. This register shows the current lock state for the Flash blocks. Each time a password group is unlocked the register becomes writable. For additional details of the LOCK bits, refer to the Flash memory description.

The LOCK bits do not come into effect before the life cycle has matured to *OEM Production* or older.

LOCK0\_PGn register functions are as shown in this section.

NOTE

Lock registers can only be accessed with 32-bit operations.

The initial value of the lock bits is set via DCF records stored in UTEST Flash block and copied over during functional reset. The default value before any DCF record is written is '1', indicating that a block is locked.

During the Customer Delivery life cycle, the user must program a valid password or a DCF record that writes to this register to unprotect the UTEST for each password group in order to prevent permanent and irreversible locking of Flash blocks.

## Warning

Failure to program valid passwords or DCF records for all password groups before the life cycle matures to OEM Production renders the part unusable.

Once a password group is unlocked, software can set or clear any of the lock bits in the group, by writing to the registers.

The resulting lock status of a Flash block is determined by the combination of the lock bits of all password groups. If a block is locked in multiple groups, then all lock bits for the block need to be cleared before program and erase is possible. However, unless the life cycle is OEM Production or older, the Flash blocks are always unlocked, regardless of the values in the password groups.

Address: 0h base + 100h offset + (16d × i), where i=0d to 3d

|       |         |          |         |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---------|----------|---------|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0       | 1        | 2       | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 |
| R     | TSLOCK  | Reserved | LOWLOCK |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |         |          |         |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | *       |          | *       | *  | *  | *  | *  | *  | *  | *  | *  | *  | *  | *  | *  | *  |
| Bit   | 16      | 17       | 18      | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | MIDLOCK |          |         |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |         |          |         |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | *       | *        | *       | *  | *  | *  | *  | *  | *  | *  | *  | *  | *  | *  | *  | *  |

\* Notes:

- MIDLOCK field: For each implemented bit of this field, if no DCF records have been written to change the lock state, the bit will reset to '1'.
- LOWLOCK field: For each implemented bit of this field, if no DCF records have been written to change the lock state, the bit will reset to '1'.
- TSLOCK field: If no DCF records have been written to change the TS bit state, the bit will reset to '1'.

## PASS\_LOCK0\_PGn field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                   |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>TSLOCK      | UTest NVM Lock. This bit is used to lock the UTest NVM block from programs (erase protection not needed since UTest NVM is OTP and not erasable). These bit fields are chip-specific and user should refer to PASS chip-specific chapter to see the flash memory regions to which they apply. |
| 1<br>Reserved    | This field is reserved.                                                                                                                                                                                                                                                                       |
| 2–15<br>LOWLOCK  | Low Block Lock. A value of 1 in a bit of the lock register signifies that the corresponding block is locked for program and erase. These bit fields are chip-specific and user should refer to PASS chip-specific chapter to see the flash memory regions to which they apply.                |
| 16–31<br>MIDLOCK | Mid Block Lock. A value of 1 in a bit of the lock register signifies that the corresponding block is locked for program and erase. These bit fields are chip-specific and user should refer to PASS chip-specific chapter to see the flash memory regions to which they apply.                |

60.3.6 Password Group *n* - Lock 1 Status Register (PASS\_LOCK1\_PG*n*)

Each Password Group has one Lock 1 Status Register. This register shows the current lock state for the Flash blocks. Each time a password group is unlocked, the register becomes writable . For additional details of the LOCK bits, refer to the Flash description.

The LOCK bits do not come into effect before the life cycle matures to OEM Production or older.

LOCK1\_PG*n* register functions are as shown in this section.

NOTE

Lock registers can only be accessed with 32-bit operations.

The initial value of the lock bits is set via DCF records stored in UTEST Flash block and copied over during functional reset . The default value before any DCF record is written is '1', which indicates a block is locked. If no DCF records have been written to change the lock state, the bits will reset to '1'.

During the Customer Delivery life cycle, the user must program a valid password or a DCF record that writes this register to unprotect the UTEST for each password group to prevent permanent and irreversible locking of Flash blocks.

CAUTION

Failure to program valid passwords or DCF records for all password groups before the life cycle matures to OEM Production renders the part unusable. However, additional DCF Records can be written to UTEST to change the initial value of this register.

Once a password group is unlocked, software can set or clear any of the lock bits in the group by writing to the registers.

The resulting lock status of a Flash block is determined by the combination of the lock bits of all password groups. If a block is locked in multiple groups, then all lock bits for the block need to be cleared before program and erase is possible. However, unless the life cycle is OEM Production or older, the Flash blocks are always unlocked, regardless of the values in the password groups.

Address: 0h base + 104h offset + (16d × i), where i=0d to 3d

|       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16       | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | 0 |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    | HIGHLOCK |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | *        | *  | *  | *  | *  | *  | *  | *  | *  | *  | *  | *  | *  | *  | *  | *  |

\* Notes:

- HIGHLOCK field: For each implemented bit of this field, if no DCF records have been written to change the lock state, the bit will reset to '1'.

### PASS\_LOCK1\_PG<sub>n</sub> field descriptions

| Field             | Description                                                                                                                                                                                                                                                                     |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>Reserved  | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                         |
| 16–31<br>HIGHLOCK | High Block Lock. A value of 1 in a bit of the lock register signifies that the corresponding block is locked for program and erase. These bit fields are chip-specific and user should refer to PASS chip-specific chapter to see the flash memory regions to which they apply. |

## 60.3.7 Password Group *n* - Lock 2 Status Register (PASS\_LOCK2\_PG<sub>n</sub>)

Each Password Group has one Lock 2 Status Register. This register shows the current lock state for the Flash blocks. Each time a password group is unlocked, the register becomes writable. For additional details of the LOCK bits, refer to the Flash description.

The LOCK bits do not come into effect before the life cycle is matured to OEM Production or older.

### NOTE

Lock registers can only be accessed with 32-bit operations.

LOCK2\_PG<sub>n</sub> register functions are as shown in this section.

The initial value of the lock bits is set via DCF records stored in UTEST Flash block and copied over during functional reset. The default value before any DCF record is written is '1', which indicates a block is locked.

During the Customer Delivery life cycle, the user must program a valid password or a DCF record that writes this register to unprotect the UTEST for each password group in order to prevent permanent and irreversible locking of Flash blocks.

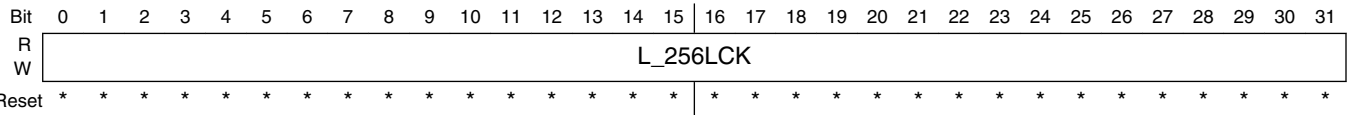
### Warning

Failure to program valid passwords or DCF records for all password groups before the life cycle matures to OEM Production renders the part unusable. However, additional DCF Records can be written to UTEST to change the initial value of this register.

Once a password group is unlocked, software can set or clear any of the lock bits in the group, by writing to the registers.

The resulting lock status of a Flash block is determined by the combination of the lock bits of all password groups. If a block is locked in multiple groups, then all lock bits for the block need to be cleared before program and erase is possible. However, unless the life cycle is OEM Production or older, the Flash blocks are always unlocked, regardless of the values in the password groups.

Address: 0h base + 108h offset + (16d × i), where i=0d to 3d



- \* Notes:
- L\_256LCK field: For each implemented bit of this field, if no DCF records have been written to change the lock state, the bit resets to '1'.

PASS\_LOCK2\_PGn field descriptions

| Field            | Description                                                                                                                                                                                                                                                                               |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>L_256LCK | Lower 256KByte Block Lock. A value of 1 in a bit of the lock register signifies that the corresponding block is locked for program and erase. These bit fields are chip-specific and user should refer to PASS chip-specific chapter to see the flash memory regions to which they apply. |

60.3.8 Password Group n - Lock 3 Status Register (PASS\_LOCK3\_PGn)

Each Password Group has one Lock 3 Status Register. This register shows the current lock state for the Flash blocks. Each time a password group is unlocked, the register becomes writable.

The LOCK bits do not come into effect before the life cycle matures to OEM Production or older.

LOCK3\_PGn register functions are as shown in this section.

NOTE

Lock registers can only be accessed with 32-bit operations.

The initial value of the lock bits is set via DCF records stored in UTEST Flash block and copied over during functional reset except for DBL and RLx bits. DBL and RLx bits are updated on a destructive reset. The default value before any DCF record is written is '1', which indicates a block is locked.

During the Customer Delivery life cycle, the user must program a valid password or a DCF record that writes this register to unprotect the UTEST for each password group in order to prevent permanent and irreversible locking of Flash blocks.



## Warning

Failure to program valid passwords or DCF records for all password groups before the life cycle matures to OEM Production renders the part unusable. However, additional DCF Records can be written to UTEST to change the initial value of this register.

Once a password group is unlocked, software can set or clear any of the lock bits in the group, by writing to the registers.

The resulting lock status of a Flash block is determined by the combination of the lock bits of all password groups. If a block is locked in multiple groups, then all lock bits for the block need to be cleared before program and erase is possible. However, unless the life cycle is OEM Production or older, the Flash blocks are always unlocked, regardless of the values in the password groups.

Once a censored device enters the Failure Analysis life cycle, read protection is active regardless whether the system is in debug mode or not. It is still possible to unlock Flash regions with the correct passwords, however. Flash Utest region controlled by LOCK3[0] is read unlocked in Failure Analysis life cycle.

## NOTE

Do not toggle DBL and RLx bits with the same write command. Use one instruction to program the DBL bit and a separate instruction to toggle the RLx bits.

Address: 0h base + 10Ch offset + (16d × i), where i=0d to 3d

| Bit   | 0   | 1   | 2  | 3 | 4    | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12  | 13  | 14  | 15  |
|-------|-----|-----|----|---|------|---|---|---|---|---|----|----|-----|-----|-----|-----|
| R     | PGL | DBL | MO | 0 | MSTR |   |   |   | 0 |   |    | 0  | RL3 | RL2 | RL1 | RL0 |
| W     |     |     |    |   |      |   |   |   |   |   |    |    |     |     |     |     |
| Reset | *   | *   | *  | 0 | 1    | 1 | 1 | 1 | 0 | 0 | 0  | *  | *   | *   | *   | *   |

| Bit   | 16       | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|-------|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| R     | U_256LCK |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | *        | *  | *  | *  | *  | *  | *  | *  | *  | *  | *  | *  | *  | *  | *  | *  |

\* Notes:

- U\_256LCK field: For each implemented bit of this field, if no DCF records have been written to change the lock state, the bit will reset to '1'.
- RL0 field: If no DCF records have been written to change the RL0 bit lock state, the bit will reset to a logic 1. This Flash block region is always readable if the life cycle is Customer Delivery or earlier. To determine if this region is readable for Life Cycle states later than Customer Delivery, consult the Flash Memory Read Protection Truth Table.
- RL1 field: If no DCF records have been written to change the RL1 bit lock state, the bit will reset to a logic 1. This Flash block region is always readable if the life cycle is Customer Delivery or earlier. To determine if this region is readable for Life Cycle states later than Customer Delivery, consult the Flash Memory Read Protection Truth Table.

## Memory map and register definition

- **RL2 field:** If no DCF records have been written to change the RL2 bit lock state, the bit will reset to a logic 1. This Flash block region is always readable if the life cycle is Customer Delivery or earlier. To determine if this region is readable for Life Cycle states later than Customer Delivery, consult the Flash Memory Read Protection Truth Table.
- **RL3 field:** If no DCF records have been written to change the RL3 bit lock state, the bit will reset to a logic 1. This Flash block region is always readable if the life cycle is Customer Delivery or earlier. To determine if this region is readable for Life Cycle states later than Customer Delivery, consult the Flash Memory Read Protection Truth Table.
- **Reserved field:** The reset value is chip-specific. See the chip-specific PASS information
- **MO field:** If no DCF records have been written to change the MO bit lock state, the bit will reset to '1'.
- **DBL field:** If no DCF records have been written to change the DBL bit lock state, the bit will reset to '1'.
- **PGL field:** Reset value depends on life cycle. If life cycle is Customer Delivery or earlier, the value will be 0, otherwise 1.

### PASS\_LOCK3\_PGn field descriptions

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0<br>PGL         | <p>Password Group Lock.</p> <p>This bit may be set in software to lock the password group, when the device is older than Customer Delivery. Before that the value of the bit is always 0.</p> <p>This bit is only updatable from SW path. It cannot be updated from DCF load from Flash.</p> <p>This bit would be automatically locked once Flash Scanning is finished after functional reset.</p> <p>This bit can be cleared by writing a valid password to the Password Challenge Input Registers.</p> <p>0 Password group registers are unlocked. All four registers in the Password group may be read and written without restriction.</p> <p>1 Password group registers are locked. Writes to the password group registers have no effect. Read accesses work normally. (If life cycle is Customer Delivery or earlier, this bit has no effect and the registers are always unlocked.)</p> |
| 1<br>DBL         | <p>Debug Interface Lock.</p> <p>The default value of this bit would be '1' if not updated from flash.</p> <p>At power on, this bit would be updated from DCF records in flash. This bit cannot be updated from FLASH after first functional reset. After first functional reset, it can only be updated by writing to it once the passgroup has been unlocked via the PASS Password challenge mechanism.</p> <p>0 Debug interface is unlocked. Accessed to all JTAG clients is allowed</p> <p>1 Debug interface is locked. Only access to JTAG password challenge register is possible. Access to all other JTAG clients is blocked</p>                                                                                                                                                                                                                                                         |
| 2<br>MO          | <p>Master Only.</p> <p>0 All masters can modify the passgroup settings if PGL is cleared</p> <p>1 Only the Master which unlocked the passgroup can change the passgroup settings</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 3<br>Reserved    | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 4–7<br>MSTR      | <p>Master Access.</p> <p>Only the Master with this matching master ID can change the passgroup settings. This field is loaded with the ID of the master which unlocked the passgroup once the correct password is provided.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 8–10<br>Reserved | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 11<br>Reserved   | <p>This field is reserved.</p> <p>This read-only field is reserved and always has the value 0.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

*Table continues on the next page...*

**PASS\_LOCK3\_PGn field descriptions (continued)**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 12<br>RL3         | <p>Region 3 is protected.</p> <p>The default value of this bit would be '1' if not updated from flash. At power on, this bit would be updated from records loaded in flash. This bit cannot be updated from FLASH after first functional reset. After first functional reset, It can only be updated by Test Access Port or Through PASS Password challenge mechanism. These bit fields are chip-specific and user should refer to PASS chip-specific chapter to see the flash memory regions to which they apply.</p> <p>0 Region 3 is not protected.<br/>1 Region 3 is protected.</p>      |
| 13<br>RL2         | <p>Region 2 is protected.</p> <p>The default value of this bit would be '1' if not updated from flash. At power on, this bit would be updated from records loaded in flash. This bit cannot be updated from FLASH after first functional reset. After first functional reset, It can only be updated by Test Access Port or Through PASS Password challenge mechanism. These bit fields are chip-specific and that user should refer to PASS chip-specific chapter to see the flash memory regions to which they apply.</p> <p>0 Region 2 is not protected.<br/>1 Region 2 is protected.</p> |
| 14<br>RL1         | <p>Region 1 is protected.</p> <p>The default value of this bit would be '1' if not updated from flash. At power on, this bit would be updated from records loaded in flash. This bit cannot be updated from FLASH after first functional reset. After first functional reset, It can only be updated by Test Access Port or Through PASS Password challenge mechanism. These bit fields are chip-specific and user should refer to PASS chip-specific chapter to see the flash memory regions to which they apply.</p> <p>0 Region 1 is not protected.<br/>1 Region 1 is protected.</p>      |
| 15<br>RL0         | <p>Region 0 is protected.</p> <p>The default value of this bit would be '1' if not updated from flash. At power on, this bit would be updated from records loaded in flash. This bit cannot be updated from FLASH after first functional reset. After first functional reset, It can only be updated by Test Access Port or Through PASS Password challenge mechanism. These bit fields are chip-specific and user should refer to PASS chip-specific chapter to see the flash memory regions to which they apply</p> <p>0 Region 0 is not protected.<br/>1 Region 0 is protected.</p>       |
| 16–31<br>U_256LCK | <p>Upper 256 KB Block Lock. A value of 1 in a bit of the lock register signifies that the corresponding block is locked for program and erase. These bit fields are chip-specific and user should refer to PASS chip-specific chapter to see the flash memory regions to which they apply</p>                                                                                                                                                                                                                                                                                                |

## 60.4 DCF clients

The following table shows the DCF clients for the PASS.

Where a DCF client is listed, the reset value of that register may automatically be pre-loaded during reset based on the data stored in DCF Records in Flash UTEST region.

The address of the DCF client is local to the PASS module and must be qualified with the appropriate Client Select (CS[14:0]) for the PASS module, as defined in the DCF Record Section of the Reference Manual or in DCF sheet attached to this document.

**Table 60-36. PASS DCF clients**

| Offset                    | Register           | Access | Reset Value | DCF Client Address (ADDR[14:0]) | Section            |
|---------------------------|--------------------|--------|-------------|---------------------------------|--------------------|
| <b>Censorship</b>         |                    |        |             |                                 |                    |
| None                      | Censorship         | None   | 0x0000_0000 | 0x00B0                          | Censorship         |
| <b>Production Disable</b> |                    |        |             |                                 |                    |
| None                      | Production Disable | None   | 0x0000_0000 | 0x00C0                          | Production Disable |

## 60.4.1 Censorship

Censorship is a DCF client that is *not* accessible by software. It is automatically pre-loaded during reset with data stored in DCF Record format in the Flash UTEST region.

|       |                |                |                |                |                |                |                |                |                |                |                |                |                |                |                |                |
|-------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
|       | 0              | 1              | 2              | 3              | 4              | 5              | 6              | 7              | 8              | 9              | 10             | 11             | 12             | 13             | 14             | 15             |
| R     |                |                |                |                |                |                |                |                |                |                |                |                |                |                |                |                |
| W     |                |                |                |                |                |                |                |                |                |                |                |                |                |                |                |                |
| Reset | x <sup>1</sup> | x <sup>1</sup> | x <sup>1</sup> | x <sup>1</sup> | x <sup>1</sup> | x <sup>1</sup> | x <sup>1</sup> | x <sup>1</sup> | x <sup>1</sup> | x <sup>1</sup> | x <sup>1</sup> | x <sup>1</sup> | x <sup>1</sup> | x <sup>1</sup> | x <sup>1</sup> | x <sup>1</sup> |

|       |                |                |                |                |                |                |                |                |                |                |                |                |                |                |                |                |
|-------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
|       | 16             | 17             | 18             | 19             | 20             | 21             | 22             | 23             | 24             | 25             | 26             | 27             | 28             | 29             | 30             | 31             |
| R     |                |                |                |                |                |                |                |                |                |                |                |                |                |                |                |                |
| W     | x <sup>2</sup> | x <sup>2</sup> | x <sup>2</sup> | x <sup>2</sup> | x <sup>2</sup> | x <sup>2</sup> | x <sup>2</sup> | x <sup>2</sup> | x <sup>2</sup> | x <sup>2</sup> | x <sup>2</sup> | x <sup>2</sup> | x <sup>2</sup> | x <sup>2</sup> | x <sup>2</sup> | x <sup>2</sup> |
| Reset | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              |

<sup>1</sup> This bit is not defined

<sup>2</sup> This value is loaded by DCF Record during reset

| Field | Description                                                                                                                       |
|-------|-----------------------------------------------------------------------------------------------------------------------------------|
| 16:31 | Censorship<br>If this value is 0x55AA the MCU is NOT censored<br>If this value is anything other than 0x55AA, the MCU is censored |

## 60.4.2 Production Disable

Production Disable is a DCF client that is not accessible by software. It is automatically pre-loaded during reset with data stored in DCF Record format in the Flash UTEST region. The Production disable DCF client defines the Debug Disable aspects of the Production Disable feature.

The DCF client implements 1 bit with a default value of '0'. The DCF client can only be written once and writes from subsequent DCF records are ignored.

For more information on Production disable, see [Production disable](#)

DCF Client Address: 0x00C0 Production Disable

Access: DCF Record Write Once Only

|       | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|-------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| R     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| W     |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| Reset | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  |

|       | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31              |
|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----------------|
| R     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                 |
| W     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | DD <sup>1</sup> |
| Reset | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0               |

<sup>1</sup> This value is loaded by DCF Record during reset

**Figure 60-34. Production Disable DCF client**

**Table 60-37. Production Disable DCF Client field descriptions**

| Field | Description                                                                                                                                                                      |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0-30  | This bit is reserved and always has the value 0.                                                                                                                                 |
| 31    | Debug disable.                                                                                                                                                                   |
| DD    | '0' = Debug Interface is Enabled and Disabled using Censorship, JTAG password and PASS_LOCK3[DBL] bit (see chip-specific PASS information).<br>'1' = Debug Interface is disabled |

### 60.4.2.1 Debug Disable (DD)

Debug Disable is an optional feature that, once the life cycle is In Field overrides all other debug controls and disables the debug interface. The Debug Disable feature has a higher priority than Censorship, JTAG password and the PASS LOCK3[DBL] bit.

## 60.5 Functional description

The primary purpose of the PASS is to provide fine-grained access control to the Flash and debugger interface. It utilizes the DCF mechanism, as described in the SSCM section.

### 60.5.1 Valid password

Each password should have at least one 1 and one 0. Other than all 0's and all 1's, all passwords are valid. If user sets all 0's or all 1's as password, then password check will always fail.

### 60.5.2 Censorship

Censorship is the overall control that enables and disables two independent security features: Debug Interface access and flash memory read protection.

The Debug Interface access is controlled by the CNS and LIFE bits of the LCSTAT register ([Life Cycle Status Register \(PASS\\_LCSTAT\)](#)) according to the Debug Interface Access truth table. The flash memory read protection is also controlled by the CNS and LIFE bits of the LCSTAT register according to the Flash memory read protection truth table.

The CNS bit is set or cleared by executing a DCF record that writes to the Censorship DCF client (See [Censorship](#)). The LIFE bits of the LCSTAT register are controlled by the SSCM using user programmed values from specific locations in the UTEST flash.

### 60.5.3 Debug Interface Access

The debug interface is fully enabled in Customer Delivery life cycle. But from OEM Production life cycle, it can be blocked using censorship and unlocked by either entering a 256 bits password via JTAG or by software clearing the DBL bit the PASS\_LOCK3 registers (See [Password Group \*n\* - Lock 3 Status Register \(PASS\\_LOCK3\\_PG\*n\*\)](#)). Both these debug enabling options are further conditioned by life cycle and Production Disable.

The debug interface enable truth table for this chip is shown in the device specific PASS section of this chapter.

### 60.5.3.1 JTAG Password Challenge

To enable the Debug Interface Access, the JTAG password challenge register in the JTAGC, needs to be written by an external tool. If that password matches the JTAG password stored in UTEST flash, Debug Interface Access is enabled until the next destructive reset.

### 60.5.4 Flash Memory Read Protection

Flash Memory Read Protection is a security feature that can selectively make regions of the flash memory unreadable. Any attempt to read a region of flash that is read protected results in the read access terminating with a transfer error. This is consistent for all bus masters including debug bus masters.

The assertion of flash memory region read protection is controlled by Life Cycle, Censorship and whether the Debug Interface access is enabled or blocked. The Flash memory read protection truth table describes the combination of conditions that drive Flash Memory Read Protection.

The intention of the feature is to protect the contents of the flash from being read. The flash memory is readable in normal operation, allowing the application to run, but read protection becomes enabled when DBL bit is cleared 0(debug interface is enabled).

The flash memory is also read protected when the life cycle status is Failure Analysis. This life cycle is used for a module field failure, where the device is returned for failure analysis. This protection prevents the contents of the flash memory from being read.

The flash memory is divided into a number of Regions (3 to 5 depending on the device). The assignment of the regions is defined in the device reference manual.

#### NOTE

When read protection is enabled all reads ( instruction fetches and data reads) are disabled and any code execution is no more possible from flash.

The memory read protection truth table for this chip is shown in the device specific PASS information.

## 60.5.5 Flash Memory Write and Erase Protection

This section describes write and erase permissions for flash memory. From power ON till system configuration and monitor block is done with system initialization, all regions of flash memory are in locked state. After power ON, System Configuration and monitor block scans flash memory for valid DCF records. These DCF records carry program and erase protection settings for flash memory regions. System Configuration and monitor block finds the settings and load them into PASS configuration via the DCF path. Once system configuration and monitor block is done with system initialization, the new configuration loaded into PASS come into force and unlocks different regions of flash for program and erase.

To enable write and erase access of a particular region of flash memory, add corresponding DCF records in the flash memory once chip comes out of reset. User can also change the settings by providing valid password to the PASS module, and by modifying the settings of following registers bits fields:

- PASS\_LOCK0\_PGn[MIDLOCK]
- PASS\_LOCK0\_PGn[LOWLOCK]
- PASS\_LOCK1\_PGn[HIGHLOCK]
- PASS\_LOCK2\_PGn[L\_256LCK]
- PASS\_LOCK3\_PGn[U\_256LCK]

See chip-specific PASS information for mapping of the above mentioned bits fields.

In the PASS module there is no individual access protection for 'write' or 'erase' access types, but has combined 'write' and 'erase' accesses.

### NOTE

This feature is controlled by the LC of the device and will become active from OEM LC on.

## 60.5.6 Production Disable

### 60.5.6.1 Overview and rationale

Production Disable is an optional feature that, once enabled, allows the MCU to function completely normally without any constraints, except the Debug interface is disabled.



### 60.5.6.2 Enabling

The feature is enabled via the Production Disable DCF client in the PASS module. Bit 0 enables the Debug Interface Disable. See [Production Disable](#) for details.

This information can be written by adding a DCF Record to the DCF record list in Flash UTEST. The DCF Client is configured to accept only one write. Subsequent DCF records that write to this DCF client are ignored.

If no DCF Record writes to this client, the default state of this feature is disabled. If, for a particular application, there is no intention of ever using this feature, it would be good practice to add a DCF Record that writes this client to 0x0000\_0000. This is not essential, but ensures the feature can not be enabled accidentally.

## 60.6 Initialization/application information

### 60.6.1 Reset

The reset state of each individual bit is shown within the Register Description section. However, many bits receive their values based on DCF records stored in the UTEST Flash memory block.

### 60.6.2 Setting lock bits in a password group

There is no restriction on setting and clearing lock registers in the UTEST Flash, apart from the available space for DCF records. It is possible to set and clear them multiple times.

**Table 60-38. CS/ADDR Values for Lock Registers**

| Register   | DCF Command Word          |
|------------|---------------------------|
| PW Group 0 |                           |
| Lock0      | 0xxxxx <sup>1</sup> _0100 |
| Lock1      | 0xxxxx <sup>1</sup> _0104 |
| Lock2      | 0xxxxx <sup>1</sup> _0108 |
| Lock3      | 0xxxxx <sup>1</sup> _010C |
| PW Group 1 |                           |
| Lock0      | 0xxxxx <sup>1</sup> _0110 |

*Table continues on the next page...*

**Table 60-38. CS/ADDR Values for Lock Registers (continued)**

| Register   | DCF Command Word          |
|------------|---------------------------|
| Lock1      | 0xxxxx <sup>1</sup> _0114 |
| Lock2      | 0xxxxx <sup>1</sup> _0118 |
| Lock3      | 0xxxxx <sup>1</sup> _011C |
| PW Group 2 |                           |
| Lock0      | 0xxxxx <sup>1</sup> _0120 |
| Lock1      | 0xxxxx <sup>1</sup> _0124 |
| Lock2      | 0xxxxx <sup>1</sup> _0128 |
| Lock3      | 0xxxxx <sup>1</sup> _012C |
| PW Group3  |                           |
| Lock0      | 0xxxxx <sup>1</sup> _0130 |
| Lock1      | 0xxxxx <sup>1</sup> _0134 |
| Lock2      | 0xxxxx <sup>1</sup> _0138 |
| Lock3      | 0xxxxx <sup>1</sup> _013C |

1. Value depends on the DCF Client Select assigned to the PASS module in the device. See chip specific PASS information for the Client Select assignment

In order to set the MIDLOCK field in the Lock0 register of password group 1 to the value 0x00FF and other fields to 0, the DCF record has to be added as shown in [Table 60-39](#).

**Table 60-39. Setting some bits in the MIDLOCK field**

| ADDR offset | DATA        | Description                              |
|-------------|-------------|------------------------------------------|
| 0x00        | 0x05AA_55AF | Start Record                             |
| 0x04        | 0x0000_0000 |                                          |
| 0x08        | 0x0000_00FF | PW Group 1, Lock0<br>Set MIDLOCK to 0xFF |
| 0x0C        | 0xxxxx_0110 |                                          |
| 0x10        | 0xFFFF_FFFF | Non-Programmed Addresses                 |
| 0x14        | 0xFFFF_FFFF |                                          |
| ...         | ...         |                                          |

# Chapter 61

## Tamper Detection Module (TDM)

### 61.1 Chip-specific TDM information

This chip has one instance of the TDM. The TDM and its associated DCF clients enable the user to:

- Select a flash memory block for tamper detection protection by assigning the block to a Tamper Detection Region (TDR)
- Override a block's tamper detection protection under certain circumstances
- Disable the override of a block's tamper detection protection
- Configure a flash memory block as One Time Programmable (OTP), meaning the block can be written only once

#### CAUTION

Specifying flash memory blocks to be monitored by the TDM and configuring flash memory blocks as OTP is done by defining DCF records. Use caution when assigning flash memory blocks to TDRs or configuring blocks as OTP. Some DCF records can be written only one time. Many DCF records can be written more than once, but the setting defined in each write is irreversible. For details about TDM-related DCF records and DCF clients, see [DCF clients](#). For additional information, see the chip-level [Device Configuration Format \(DCF\) Records](#) details.

#### 61.1.1 Disabling the override of tamper detection

The [Tamper Region Override \(TO\) DCF client](#) allows the user to override tamper detection protection. The override cancels tamper detection protection for all blocks in a TDR and is irreversible.

The [Software Tamper Region Override Disable \(STO\\_DIS\) DCF client](#) allows the user to disable the override option permanently.

To optimize security on this chip, disable the Tamper Region Override option permanently by using the Software Tamper Region Override Disable DCF client.

## 61.1.2 TDR<sub>n</sub>\_LOCK<sub>m</sub> DCF client bit mapping

The TDR<sub>n</sub>\_LOCK<sub>m</sub> DCF clients provide a means to select flash memory blocks for inclusion in a TDR. Specific memory blocks map to particular bits in TDR<sub>n</sub>\_LOCK<sub>m</sub> clients.

### 61.1.2.1 TDR<sub>n</sub>\_LOCK0 DCF client bit mapping

The flash memory blocks mapped to the [Tamper Region Assignment DCF client \(TDR<sub>n</sub>\\_LOCK0\)](#) bits are as follows.

|                                        | 0        | 1        | 2        | 3        | 4               | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15              | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|----------------------------------------|----------|----------|----------|----------|-----------------|---|---|---|---|---|----|----|----|----|----|-----------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Register Field                         | Reserved | Reserved | Reserved | Reserved | LOWTDRAGN[11:0] |   |   |   |   |   |    |    |    |    |    | MIDTDRAGN[15:0] |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Flash Memory Block Name                | —        | —        | —        | —        | —               | — | — | — | — | — | —  | —  | —  | —  | —  | —               | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  |
| 64 KB EEPROM Data Block1 - Partition 1 |          |          |          |          |                 |   |   |   |   |   |    |    |    |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 64 KB EEPROM Data Block0 - Partition 0 |          |          |          |          |                 |   |   |   |   |   |    |    |    |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 64 KB EEPROM Data Block1 - Partition 3 |          |          |          |          |                 |   |   |   |   |   |    |    |    |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 64 KB EEPROM Data Block0 - Partition 2 |          |          |          |          |                 |   |   |   |   |   |    |    |    |    |    |                 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

61.1.2.2 TDRn\_LOCK1 DCF client bit mapping

The flash memory blocks mapped to the [Tamper Region Assignment DCF client \(TDRn\\_LOCK1\)](#) bits are as follows.

NOTE

Because the CSE module has exclusive access to these flash memory blocks, this DCF client and the associated DCF records are ignored.

|                         |          | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15               | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31                                  |                                     |
|-------------------------|----------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------------------------------|-------------------------------------|
| Register Field          | Reserved |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    | HIGHTDRAGN[15:0] |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                                     |                                     |
| Flash Memory Block Name | —        | — | — | — | — | — | — | — | — | — | — | —  | —  | —  | —  | —  | —                | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | 16 KB CSE Data Block1 - Partition 5 | 16 KB CSE Data Block0 - Partition 4 |

61.1.2.3 TDRn\_LOCK2 DCF client bit mapping

The flash memory blocks mapped to the [Tamper Region Assignment DCF client \(TDRn\\_LOCK2\)](#) bits are as follows.

|                | 0                 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|----------------|-------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Register Field | 256KBTDRAIN[31:0] |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

Table continues on the next page...

| Flash Memory Block Name                       |
|-----------------------------------------------|
| 256 KB Application Code Block31 - Partition 9 |
| 256 KB Application Code Block30 - Partition 9 |
| 256 KB Application Code Block29 - Partition 9 |
| 256 KB Application Code Block28 - Partition 9 |
| 256 KB Application Code Block27 - Partition 9 |
| 256 KB Application Code Block26 - Partition 9 |
| 256 KB Application Code Block25 - Partition 9 |
| 256 KB Application Code Block24 - Partition 9 |
| 256 KB Application Code Block23 - Partition 8 |
| 256 KB Application Code Block22 - Partition 8 |
| 256 KB Application Code Block21 - Partition 8 |
| 256 KB Application Code Block20 - Partition 8 |
| 256 KB Application Code Block19 - Partition 8 |
| 256 KB Application Code Block18 - Partition 8 |
| 256 KB Application Code Block17 - Partition 8 |
| 256 KB Application Code Block16 - Partition 8 |
| 256 KB Application Code Block15 - Partition 7 |
| 256 KB Application Code Block14 - Partition 7 |
| 256 KB Application Code Block13 - Partition 7 |
| 256 KB Application Code Block12 - Partition 7 |
| 256 KB Application Code Block11 - Partition 7 |
| 256 KB Application Code Block10 - Partition 7 |
| 256 KB Application Code Block9 - Partition 7  |
| 256 KB Application Code Block8 - Partition 7  |
| 256 KB Application Code Block7 - Partition 6  |
| 256 KB Application Code Block6 - Partition 6  |
| 256 KB Application Code Block5 - Partition 6  |
| Calibration Block4 - Partition 6              |
| Calibration Block3 - Partition 6              |
| Calibration Block2 - Partition 6              |
| Calibration Block1 - Partition 6              |
| Boot Code Block0 - Partition 6                |

### 61.1.2.4 TDR<sub>n</sub>\_LOCK3 DCF client bit mapping

The flash memory blocks mapped to the [Tamper Region Assignment DCF client \(TDR<sub>n</sub>\\_LOCK3\)](#) bits are as follows.

|                         | 0        | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15                | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|-------------------------|----------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|-------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Register Field          | Reserved |   |   |   |   |   |   |   |   |   |    |    |    |    |    | 256KBTDRAgn[15:0] |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Flash Memory Block Name |          |   |   |   |   |   |   |   |   |   |    |    |    |    |    |                   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

### 61.1.3 OTPEN<sub>n</sub> DCF client bit mapping

The OTPEN<sub>n</sub> DCF clients provide a means to configure flash memory blocks to be One Time Programmable (OTP). Specific memory blocks map to particular bits in OTPEN<sub>n</sub> clients.

#### 61.1.3.1 OTPEN<sub>0</sub> DCF client bit mapping

The flash memory blocks mapped to the [One Time Programmable Enable \(OTPEN<sub>0</sub>\) DCF client](#) bits are as follows.

|                                        | 0        | 1        | 2        | 3        | 4              | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15             | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|----------------------------------------|----------|----------|----------|----------|----------------|---|---|---|---|---|----|----|----|----|----|----------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Register Field                         | Reserved | Reserved | Reserved | Reserved | LOWOTPEN[11:0] |   |   |   |   |   |    |    |    |    |    | MIDOTPEN[15:0] |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Flash Memory Block Name                | —        | —        | —        | —        | —              | — | — | — | — | — | —  | —  | —  | —  | —  | —              | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  |
| 64 KB EEPROM Data Block1 - Partition 1 |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 64 KB EEPROM Data Block0 - Partition 0 |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| —                                      |          |          |          |          |                |   |   |   |   |   |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

#### 61.1.3.2 OTPEN<sub>1</sub> DCF client bit mapping

The flash memory blocks mapped to the [One Time Programmable Enable \(OTPEN<sub>1</sub>\) DCF client](#) bits are as follows.

#### NOTE

When the value of a bit in this DCF client reflects that the corresponding flash block is one-time programmable (OTP),

Chip-specific TDM information

the OTP status applies to the block even though the CSE module has exclusive access to the block.

|                         | 0        | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15              | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31                                  |                                     |
|-------------------------|----------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|-----------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------------------------------------|-------------------------------------|
| Register Field          | Reserved |   |   |   |   |   |   |   |   |   |    |    |    |    |    | HIGHOTPEN[15:0] |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                                     |                                     |
| Flash Memory Block Name | —        | — | — | — | — | — | — | — | — | — | —  | —  | —  | —  | —  | —               | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | —  | 16 KB CSE Data Block1 - Partition 5 | 16 KB CSE Data Block0 - Partition 4 |

61.1.3.3 OTPEN2 DCF client bit mapping

The flash memory blocks mapped to the [One Time Programmable Enable \(OTPEN2\) DCF client](#) bits are as follows.

|                | 0                | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|----------------|------------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Register Field | 256KBOTPEN[31:0] |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |

Table continues on the next page...



| Flash Memory Block Name                       |
|-----------------------------------------------|
| 256 KB Application Code Block31 - Partition 9 |
| 256 KB Application Code Block30 - Partition 9 |
| 256 KB Application Code Block29 - Partition 9 |
| 256 KB Application Code Block28 - Partition 9 |
| 256 KB Application Code Block27 - Partition 9 |
| 256 KB Application Code Block26 - Partition 9 |
| 256 KB Application Code Block25 - Partition 9 |
| 256 KB Application Code Block24 - Partition 9 |
| 256 KB Application Code Block23 - Partition 8 |
| 256 KB Application Code Block22 - Partition 8 |
| 256 KB Application Code Block21 - Partition 8 |
| 256 KB Application Code Block20 - Partition 8 |
| 256 KB Application Code Block19 - Partition 8 |
| 256 KB Application Code Block18 - Partition 8 |
| 256 KB Application Code Block17 - Partition 8 |
| 256 KB Application Code Block16 - Partition 8 |
| 256 KB Application Code Block15 - Partition 7 |
| 256 KB Application Code Block14 - Partition 7 |
| 256 KB Application Code Block13 - Partition 7 |
| 256 KB Application Code Block12 - Partition 7 |
| 256 KB Application Code Block11 - Partition 7 |
| 256 KB Application Code Block10 - Partition 7 |
| 256 KB Application Code Block9 - Partition 7  |
| 256 KB Application Code Block8 - Partition 7  |
| 256 KB Application Code Block7 - Partition 6  |
| 256 KB Application Code Block6 - Partition 6  |
| 256 KB Application Code Block5 - Partition 6  |
| Calibration Block4 - Partition 6              |
| Calibration Block3 - Partition 6              |
| Calibration Block2 - Partition 6              |
| Calibration Block1 - Partition 6              |
| Boot Code Block0 - Partition 6                |

### 61.1.3.4 OTPEN3 DCF client bit mapping

The flash memory blocks mapped to the [One Time Programmable Enable \(OTPEN3\) DCF client](#) bits are as follows.

| 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 |          |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|---------------------------------------------------------------------------------------|----------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|------------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| Register Field                                                                        | Reserved |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 256KBOTPEN[15:0] |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Flash Memory Block Name                                                               |          |  |  |  |  |  |  |  |  |  |  |  |  |  |  |                  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

## 61.2 Overview

The Tamper Detection Module provides a type of flash memory erase protection mechanism that requires software to write a record associated with one or more blocks in a Tamper Detection Region (TDR) before the block(s) can be erased. The mechanism does not *prohibit* an erase operation on any block within a TDR—it requires a record to be written before the operation can execute. Collectively, the records are referred to as a “diary”. At minimum, the diary serves as an erase counter. Because the diary record content is customer-defined, the diary can also serve as a history.

Up to 6 TDRs can be defined via DCF records.

The following figure shows the block diagram of the TDM and its interface to other system components.

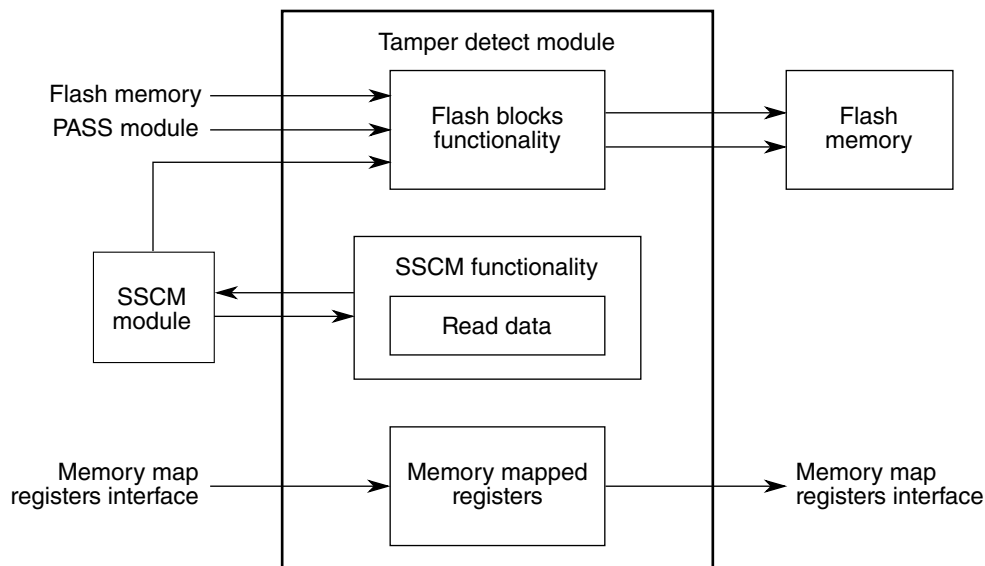


**Figure 61-1. Tamper Detection Module block diagram**

Security information is passed to the TDM by the SSCM module via the DCF bus. Among the data are:

- OTP (One Time Programmable) settings for each of the flash blocks
- Base address for the diary in the flash
- Block assignments for the TDRs
- Tamper region override information
- Software tamper region disable

The TDM module controls the flash blocks for erase operations and sets them as OTP.

The two flash buses are used to control the flash blocks for erase operation and set them as OTP. OTP means that flash erase of the entire block is disabled and the only words that are already erased can be programmed. Over-programming is not possible.

### NOTE

The pass module provides the control for erase operation for each flash block as well as for each region of flash memory, and the PASS settings have priority over the settings of TDM.

The flash block indicates to the TDM the address of the programmed word. This address is internally decoded and provides a means to verify whether the block is part of any TDRs. The memory map registers interface provides a means to the user to read information of the TDM registers.

## 61.3 Features

The Tamper Detection Module supports these distinctive features:

- Erase counter (diary) for each TDR to record up to 256 write attempts on the flash blocks
  - The diary is divided into 6 parts (TDRs), each consisting of 256 64-bit records.
  - The diary block should be assigned as OTP region in flash.
  - The diary area can store any type of relevant information.
- Tamper Detection Region Block Assignment
  - Any flash block can be assigned to any TDR and be controlled.
- 32-bit Erase Lock DCF clients for each flash memory TDR defined.
- 32-bit OTP Enable DCF clients to disable the erase operation for an entire flash block
- Override TDR via DCF records<sup>1</sup>
  - Allows block assigned to a TDR to be unlocked for erase even if the TDR is locked for this operation.
- Memory mapped registers interface for reading via software of specific registers.

## 61.4 External signal description

There are no external signals driving or being driven by the TDM.

---

1. The Tamper Region Override DCF record is a "write one only" DCF record. If an override is applied to a TDR it is permanent and cannot be reversed.

## 61.5 DCF clients

TDM-related DCF records perform the following functions:

- Establish a permanent diary base address
- Define Tamper Detect Regions (TDRs) to monitor on-chip flash memory program/erase activity
- Permanently disable one or more Tamper Regions
- Permanently disable ability for software to permanently override one or more TDRs
- Configure flash memory as One Time Programmable (OTP) on a per-block basis

The following table gives an overview on the TDM DCF clients implemented.

**Table 61-1. DCF clients**

| DCF Client Name                         | Description                                                                                                                                                                                                                | Strategy                      |
|-----------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------|
| Diary Base Address                      | Stores the base address of the tamper detection diary                                                                                                                                                                      | write once, write 1 to 0 only |
| Tamper Region Override                  | Stores information of which TDR is to be overridden                                                                                                                                                                        | write 0 to 1 only             |
| Software Tamper Region Override Disable | Stores information to disable the software mechanism to override the TDR                                                                                                                                                   | write once, write 0 to 1 only |
| OTPEN[nb] (OTP Enable)                  | Stores information of which flash blocks is to be assigned as OTP (One Time Programmable) - nb means one register for each Lock register of the flash memory                                                               | write 0 to 1 only             |
| TDR[n]_LOCK[m] Region Assignment        | Stores information of which flash block is assigned to a TDR. <ul style="list-style-type: none"> <li>• n specifies a TDR</li> <li>• m specifies the flash blocks as per the LOCK registers of the flash memory.</li> </ul> | write 0 to 1 only             |

### 61.5.1 Diary Base Address (DBA) DCF client

This DCF client holds base address information of the diary. The following diagram depicts this client.

**CAUTION**

This write-once DCF client can be written only one time. After one DCF record has written to this client, all subsequent writes are ignored.

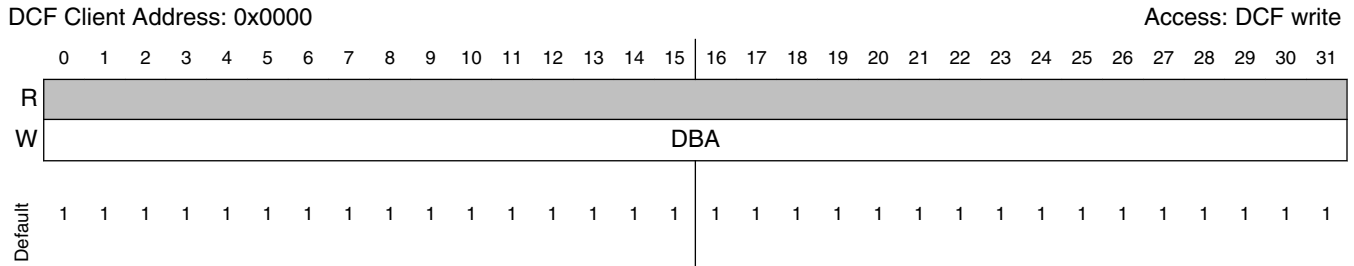


**Table 61-2. Diary Base Address (DBA) DCF client field description**

| Field       | Description                                                                                                                                                                                                                           |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>DBA | Diary Base Address. This field holds the base address of the diary. This information is then used to verify if the diary being programmed matches any TDR region.<br><br><b>NOTE:</b> Diary Base Address must be on a 16 KB boundary. |

## 61.5.2 Tamper Region Override (TO) DCF client

This DCF client holds override information for each TDR. The following figure shows the diagram for this register.

**CAUTION**

This DCF client is writable from 0 to 1 only. Attempted writes of bits from 1 to 0 are ignored, making the override of a TDR permanent.

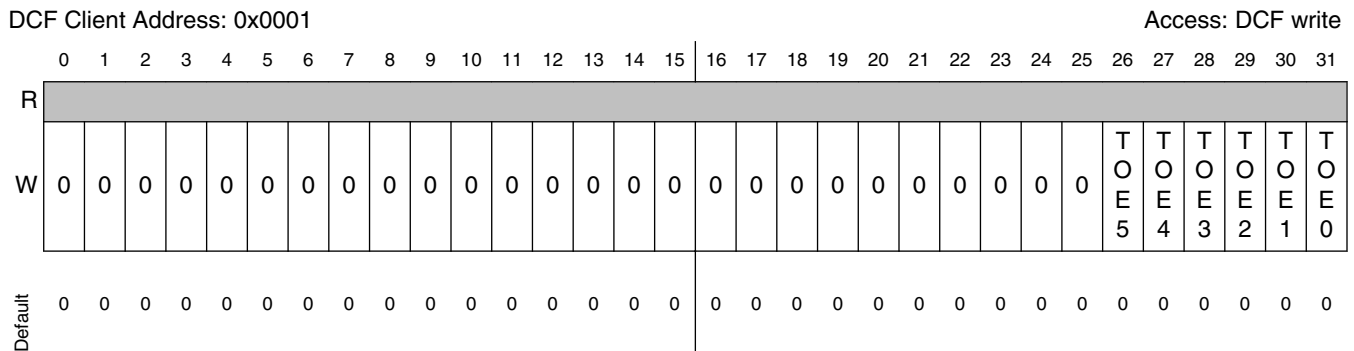


**Table 61-3. Tamper Region Override (TO) DCF client field descriptions**

| Field                 | Description                                                                                                                                                                                                                                                                                                                                                          |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–25<br>Reserved      | This field is reserved and should be written with all zeroes when programming the TOE fields.                                                                                                                                                                                                                                                                        |
| 26–31<br>TOE5 to TOE0 | TDR Override Enable. Each TOE <sub>n</sub> field, where <i>n</i> represents the TDRs, holds information of the Override Enable status for each TDR.<br><div> 0 Normal operation. Blocks assigned to this TDR can NOT be erased until the diary is written.<br/> 1 Diary override. The blocks assigned to this TDR can be erased WITHOUT writing to the diary. </div> |

### 61.5.3 Software Tamper Region Override Disable (STO\_DIS) DCF client

This DCF client holds information to disable the software mechanism to override the tamper detect regions.

#### CAUTION

This write-once DCF client can be written only one time. After one DCF record has written to this client, all subsequent writes are ignored.

DCF Client Address: 0x0002

Access: DCF write

|         |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|---------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
|         | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W       | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |
|         |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**Table 61-4. Software Tamper Region Override Disable (STO\_DIS) DCF client field descriptions**

| Field            | Description                                                                                       |
|------------------|---------------------------------------------------------------------------------------------------|
| 0–25<br>Reserved | This field is reserved and should be written with all zeroes when programming the STO_DIS fields. |

Table continues on the next page...

**Table 61-4. Software Tamper Region Override Disable (STO\_DIS) DCF client field descriptions (continued)**

| Field                         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 26–31<br>STO_DIS5 to STO_DIS0 | Software TDR Override Disable TDR <sub>n</sub> . Each STO_DIS <sub>n</sub> field, where <i>n</i> represents the TDR number, holds information to disable the software mechanism to override TDR <sub>n</sub> .<br>0 Software tamper detect override mechanism supported<br>1 Software tamper detect override mechanism disabled. Any attempt to override TDR <sub>n</sub> by loading a service key value in STO_KEY <sub>n</sub> register is ignored. |

### 61.5.4 One Time Programmable Enable (OTPEN0) DCF client

This DCF client holds OTP information for particular blocks of flash memory. This DCF client mirrors the LOCK0 register of the flash memory. The following diagram depicts this client.

#### CAUTION

This DCF client is writable from 0 to 1 only. Attempted writes of bits from 1 to 0 are ignored, making the OTP assignment of a flash memory block irreversible.

DCF Client Address: 0x0008

Access: DCF write

|         |   |   |   |   |                |   |   |   |   |   |    |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|---------|---|---|---|---|----------------|---|---|---|---|---|----|----|----|----|----|----|----------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
|         | 0 | 1 | 2 | 3 | 4              | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16             | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R       |   |   |   |   |                |   |   |   |   |   |    |    |    |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W       | 0 | 0 | 0 | 0 | LOWOTPEN[11:0] |   |   |   |   |   |    |    |    |    |    |    | MIDOTPEN[15:0] |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Default | 0 | 0 | 0 | 0 | 0              | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0              | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**Table 61-5. One Time Programmable Enable (OTPEN0) DCF client**

| Field             | Description                                                                                                                                                                                                                                                                                                                                                                  |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–3<br>Reserved   | This field is reserved and should be written with all zeroes when writing to this client.                                                                                                                                                                                                                                                                                    |
| 4–15<br>LOWOTPEN  | Low Block OTP Enable. This field affects flash memory blocks in Low address space. See the chip-specific information for the mapping between field bits and flash blocks. The same mapping also applies to control similar functions in the flash memory module.<br>0 Corresponding flash blocks are not assigned as OTP<br>1 Corresponding flash blocks are assigned as OTP |
| 16–31<br>MIDOTPEN | Mid Block OTP Enable. This field affects flash memory blocks in Mid address space. See the chip-specific information for the mapping between field bits and flash blocks. The same mapping also applies to control similar functions in the flash memory module.<br>0 Corresponding flash blocks are not assigned as OTP<br>1 Corresponding flash blocks are assigned as OTP |

61.5.5 One Time Programmable Enable (OTPEN1) DCF client

This DCF client holds OTP information for particular blocks of flash memory. This DCF client mirrors the LOCK1 register of the flash memory. The following diagram depicts the client.

CAUTION

This DCF client is writable from 0 to 1 only. Attempted writes of bits from 1 to 0 are ignored, making the OTP assignment of a flash memory block irreversible.

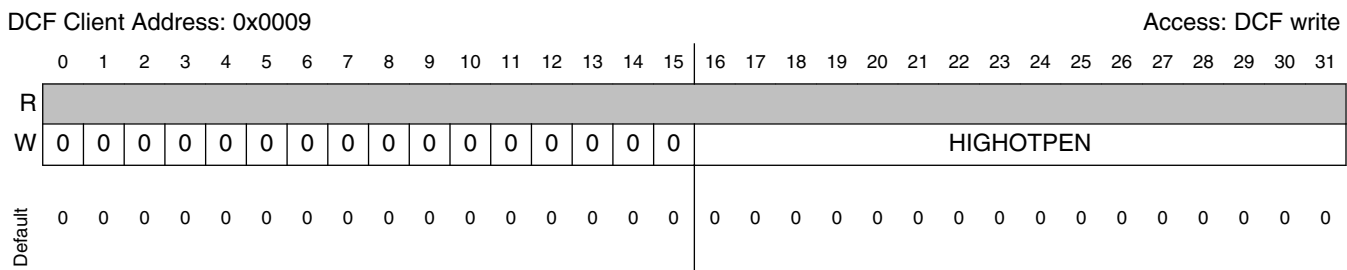


Table 61-6. One Time Programmable Enable (OTPEN1) DCF client field descriptions

| Field              | Description                                                                                                                                                                                                                                                                                                                                                                                  |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>Reserved   | This field is reserved and should be written with zeroes when writing to this client.                                                                                                                                                                                                                                                                                                        |
| 16–31<br>HIGHOTPEN | High Block OTP Enable. This field affects flash memory blocks in High address space. See the chip-specific information for the mapping between field bits and flash blocks. The same mapping also applies to control similar functions in the flash memory module.<br><br>0      Corresponding flash blocks are not assigned as OTP<br>1      Corresponding flash blocks are assigned as OTP |

61.5.6 One Time Programmable Enable (OTPEN2) DCF client

This DCF client holds OTP information for particular blocks of flash memory. This DCF client mirrors the LOCK2 register of the flash memory. The following diagram depicts the client.



**CAUTION**

This DCF client is writable from 0 to 1 only. Attempted writes of bits from 1 to 0 are ignored, making the OTP assignment of a flash memory block irreversible.

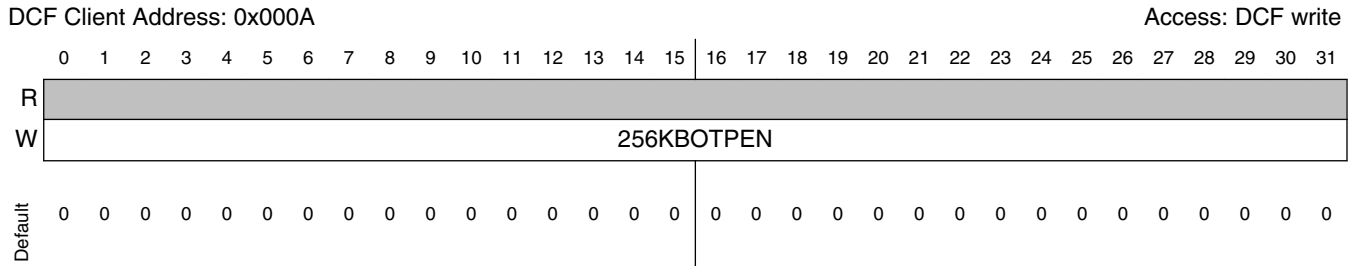


**Table 61-7. One Time Programmable Enable (OTPEN2) DCF client field descriptions**

| Field              | Description                                                                                                                                                                                                                                                                                                                                                                                      |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>256KBOTPEN | <p>256 KB Block OTP Enable. This field affects flash memory blocks in 256KB address space. See the chip-specific information for the mapping between field bits and flash blocks. The same mapping also applies to control similar functions in the flash memory module.</p> <p>0 Corresponding flash blocks are not assigned as OTP</p> <p>1 Corresponding flash blocks are assigned as OTP</p> |

### 61.5.7 One Time Programmable Enable (OTPEN3) DCF client

This DCF client holds OTP information for particular blocks of flash memory. This DCF client mirrors the LOCK3 register of the flash memory. The following diagram depicts the client.

**CAUTION**

This DCF client is writable from 0 to 1 only. Attempted writes of bits from 1 to 0 are ignored, making the OTP assignment of a flash memory block irreversible.

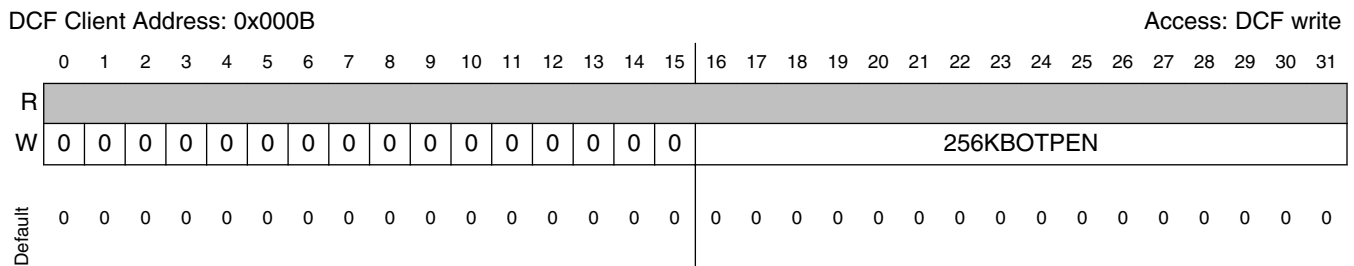


**Table 61-8. One Time Programmable Enable (OTPEN3) DCF client field descriptions**

| Field               | Description                                                                                                                                                                                                                                                                                                                                                                                          |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>Reserved    | This field is reserved and should be written with zeroes when writing to this client.                                                                                                                                                                                                                                                                                                                |
| 16–31<br>256KBOTPEN | 256 KB Block OTP Enable. This field affects flash memory blocks in the upper range of 256KB address space. See the chip-specific information for the mapping between field bits and flash blocks. The same mapping also applies to control similar functions in the flash memory module.<br>0 Corresponding flash blocks are not assigned as OTP<br>1 Corresponding flash blocks are assigned as OTP |

### 61.5.8 Tamper Region Assignment DCF client (TDRn\_LOCK0)

This DCF client holds the assignment information of flash memory blocks to a specific TDR. Each flash memory block can be assigned to any TDR ( $n$  varies from 0 to 5). This DCF client holds the assignment information for only those blocks specified in the LOCK0 register in the flash memory. For that reason, the layout of this DCF client mirrors the LOCK0 register in the flash memory. The following diagram depicts the TDRn\_LOCK0 DCF client.

#### CAUTION

This DCF client is writable from 0 to 1 only. Attempted writes of bits from 1 to 0 are ignored, making the assignment of a flash memory block to a TDR irreversible unless the entire TDR is overridden.

DCF Client Address:  $0x0014 + ((n \times 4) + 0)$ 

Access: DCF write

|         |   |   |   |   |           |   |   |   |   |   |    |    |    |    |    |    |           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |
|---------|---|---|---|---|-----------|---|---|---|---|---|----|----|----|----|----|----|-----------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--|
|         | 0 | 1 | 2 | 3 | 4         | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16        | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |  |
| R       |   |   |   |   |           |   |   |   |   |   |    |    |    |    |    |    |           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |
| W       | 0 | 0 | 0 | 0 | LOWTDRAGN |   |   |   |   |   |    |    |    |    |    |    | MIDTDRAGN |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |  |
| Default | 0 | 0 | 0 | 0 | 0         | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0         | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |  |

**Table 61-9. Tamper Region Assignment DCF client (TDRn\_LOCK0)**

| Field           | Description                                                                           |
|-----------------|---------------------------------------------------------------------------------------|
| 0–3<br>Reserved | This field is reserved and should be written with zeroes when writing to this client. |

*Table continues on the next page...*

**Table 61-9. Tamper Region Assignment DCF client (TDRn\_LOCK0) (continued)**

| Field              | Description                                                                                                                                                                                                                                                                                                                                                                                          |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 4–15<br>LOWTDRAGN  | Low Block Tamper Region Assignment. This field affects flash memory blocks in Low address space. See the chip-specific information for the mapping between field bits and flash blocks. The same mapping also applies to control similar functions in the flash memory module.<br>0 Corresponding flash blocks are not assigned to the TDRn<br>1 Corresponding flash blocks are assigned to the TDRn |
| 16–31<br>MIDTDRAGN | Mid Block Tamper Region Assignment. This field affects flash memory blocks in Mid address space. See the chip-specific information for the mapping between field bits and flash blocks. The same mapping also applies to control similar functions in the flash memory module.<br>0 Corresponding flash blocks are not assigned to the TDRn<br>1 Corresponding flash blocks are assigned to the TDRn |

### 61.5.9 Tamper Region Assignment DCF client (TDRn\_LOCK1)

This DCF client holds the assignment information of flash memory blocks to a specific TDR. Each flash memory block can be assigned to any TDR ( $n$  varies from 0 to 5). This DCF client holds the assignment information for only those blocks specified in the LOCK1 register in the flash memory. For that reason, the layout of this DCF client mirrors the LOCK1 register in the flash memory. The following diagram depicts the TDRn\_LOCK1 DCF client.

#### CAUTION

This DCF client is writable from 0 to 1 only. Attempted writes of bits from 1 to 0 are ignored, making the assignment of a flash memory block to a TDR irreversible unless the entire TDR is overridden.

DCF Client Address:  $0x0014 + ((n \times 4) + 1)$ 

Access: DCF write

|         |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|---------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
|         | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15         | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W       | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | HIGHTDRAGN |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0          | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**Table 61-10. Tamper Region Assignment DCF client (TDRn\_LOCK1)**

| Field            | Description                                                                           |
|------------------|---------------------------------------------------------------------------------------|
| 0–15<br>Reserved | This field is reserved and should be written with zeroes when writing to this client. |

*Table continues on the next page...*

**Table 61-10. Tamper Region Assignment DCF client (TDRn\_LOCK1) (continued)**

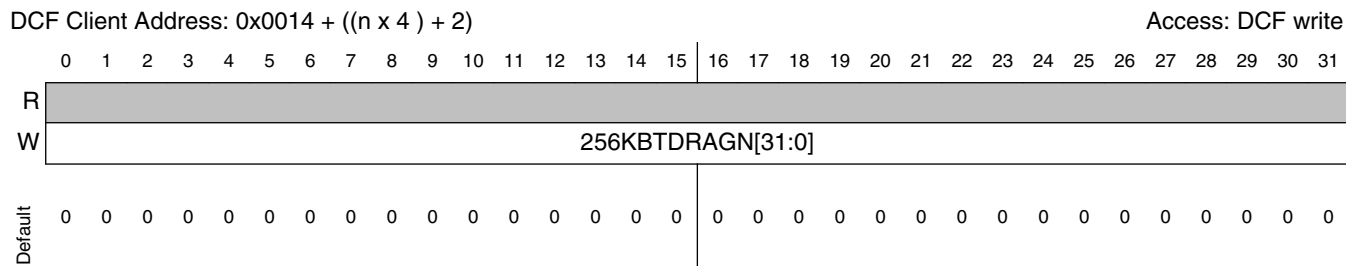
| Field               | Description                                                                                                                                                                                                                                                                      |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 16–31<br>HIGHTDRAGN | High Block Tamper Region Assignment. This field affects flash memory blocks in High address space. See the chip-specific information for the mapping between field bits and flash blocks. The same mapping also applies to control similar functions in the flash memory module. |
| 0                   | Corresponding flash blocks are not assigned to the TDRn                                                                                                                                                                                                                          |
| 1                   | Corresponding flash blocks are assigned to the TDRn                                                                                                                                                                                                                              |

### 61.5.10 Tamper Region Assignment DCF client (TDRn\_LOCK2)

This DCF client holds the assignment information of flash memory blocks to a specific TDR. Each flash memory block can be assigned to any TDR (*n* varies from 0 to 5). This DCF client holds the assignment information for only those blocks specified in the LOCK2 register in the flash memory. For that reason, the layout of this DCF client mirrors the LOCK2 register in the flash memory. The following diagram depicts the TDRn\_LOCK2 DCF client.

#### CAUTION

This DCF client is writable from 0 to 1 only. Attempted writes of bits from 1 to 0 are ignored, making the assignment of a flash memory block to a TDR irreversible unless the entire TDR is overridden.

**Table 61-11. Tamper Region Assignment DCF client (TDRn\_LOCK2)**

| Field                     | Description                                                                                                                                                                                                                                                                         |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>256KBTDRAGN[31:0] | 256 KB Block Tamper Region Assignment. This field affects flash memory blocks in 256KB address space. See the chip-specific information for the mapping between field bits and flash blocks. The same mapping also applies to control similar functions in the flash memory module. |
| 0                         | Corresponding flash blocks are not assigned to the TDRn                                                                                                                                                                                                                             |
| 1                         | Corresponding flash blocks are assigned to the TDRn                                                                                                                                                                                                                                 |

### 61.5.11 Tamper Region Assignment DCF client (TDRn\_LOCK3)

This DCF client holds the assignment information of flash memory blocks to a specific TDR. Each flash memory block can be assigned to any TDR ( $n$  varies from 0 to 5). This DCF client holds the assignment information for only those blocks specified in the LOCK3 register in the flash memory. For that reason, the layout of this DCF client mirrors the LOCK3 register in the flash memory. The following diagram depicts the TDRn\_LOCK3 DCF client.

#### CAUTION

This DCF client is writable from 0 to 1 only. Attempted writes of bits from 1 to 0 are ignored, making the assignment of a flash memory block to a TDR irreversible unless the entire TDR is overridden.

DCF Client Address:  $0x0014 + ((n \times 4) + 3)$

Access: DCF write

|         |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |                    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|---------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|--------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
|         | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15                 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |                    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W       | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 256KBTDRAgn[47:32] |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Default | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0                  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

**Table 61-12. Tamper Region Assignment DCF client (TDRn\_LOCK3)**

| Field                       | Description                                                                                                                                                                                                                                                                                                                                                                                              |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–15<br>Reserved            | This field is reserved and should be written with zeroes when writing to this client.                                                                                                                                                                                                                                                                                                                    |
| 16–31<br>256KBTDRAgn[47:32] | 256KB Block Tamper Region Assignment. This field affects flash memory blocks in 256KB address space. See the chip-specific information for the mapping between field bits and flash blocks. The same mapping also applies to control similar functions in the flash memory module.<br>0 Corresponding flash blocks are not assigned to the TDRn<br>1 Corresponding flash blocks are assigned to the TDRn |

## 61.6 Memory Map/Register Definition

#### TDM memory map

| Address offset (hex) | Register name                                      | Width (in bits) | Access | Reset value                 | Section/ page               |
|----------------------|----------------------------------------------------|-----------------|--------|-----------------------------|-----------------------------|
| 0                    | TDR Status Register (TDM_TDRSR)                    | 32              | R      | <a href="#">See section</a> | <a href="#">61.6.1/2762</a> |
| 4                    | Last Flash Programmed Address Register (TDM_LFPAR) | 32              | R      | 0000_0000h                  | <a href="#">61.6.2/2764</a> |

*Table continues on the next page...*

**TDM memory map (continued)**

| Address offset (hex) | Register name                                      | Width (in bits) | Access | Reset value                 | Section/ page               |
|----------------------|----------------------------------------------------|-----------------|--------|-----------------------------|-----------------------------|
| 8                    | Diary Base Address (TDM_DBA)                       | 32              | R      | <a href="#">See section</a> | <a href="#">61.6.3/2765</a> |
| 10                   | Software Tamper Override Key Region (TDM_STO_KEY0) | 32              | R/W    | 0000_0000h                  | <a href="#">61.6.4/2765</a> |
| 14                   | Software Tamper Override Key Region (TDM_STO_KEY1) | 32              | R/W    | 0000_0000h                  | <a href="#">61.6.4/2765</a> |
| 18                   | Software Tamper Override Key Region (TDM_STO_KEY2) | 32              | R/W    | 0000_0000h                  | <a href="#">61.6.4/2765</a> |
| 1C                   | Software Tamper Override Key Region (TDM_STO_KEY3) | 32              | R/W    | 0000_0000h                  | <a href="#">61.6.4/2765</a> |
| 20                   | Software Tamper Override Key Region (TDM_STO_KEY4) | 32              | R/W    | 0000_0000h                  | <a href="#">61.6.4/2765</a> |
| 24                   | Software Tamper Override Key Region (TDM_STO_KEY5) | 32              | R/W    | 0000_0000h                  | <a href="#">61.6.4/2765</a> |

**61.6.1 TDR Status Register (TDM\_TDRSR)**

This register contains the current status, i.e., locked or unlocked, of each tamper region, i.e., TDR.

- If a tamper region is locked, the blocks assigned to that tamper region cannot be erased.
- If a region is unlocked, the blocks within the tamper region can be erased.

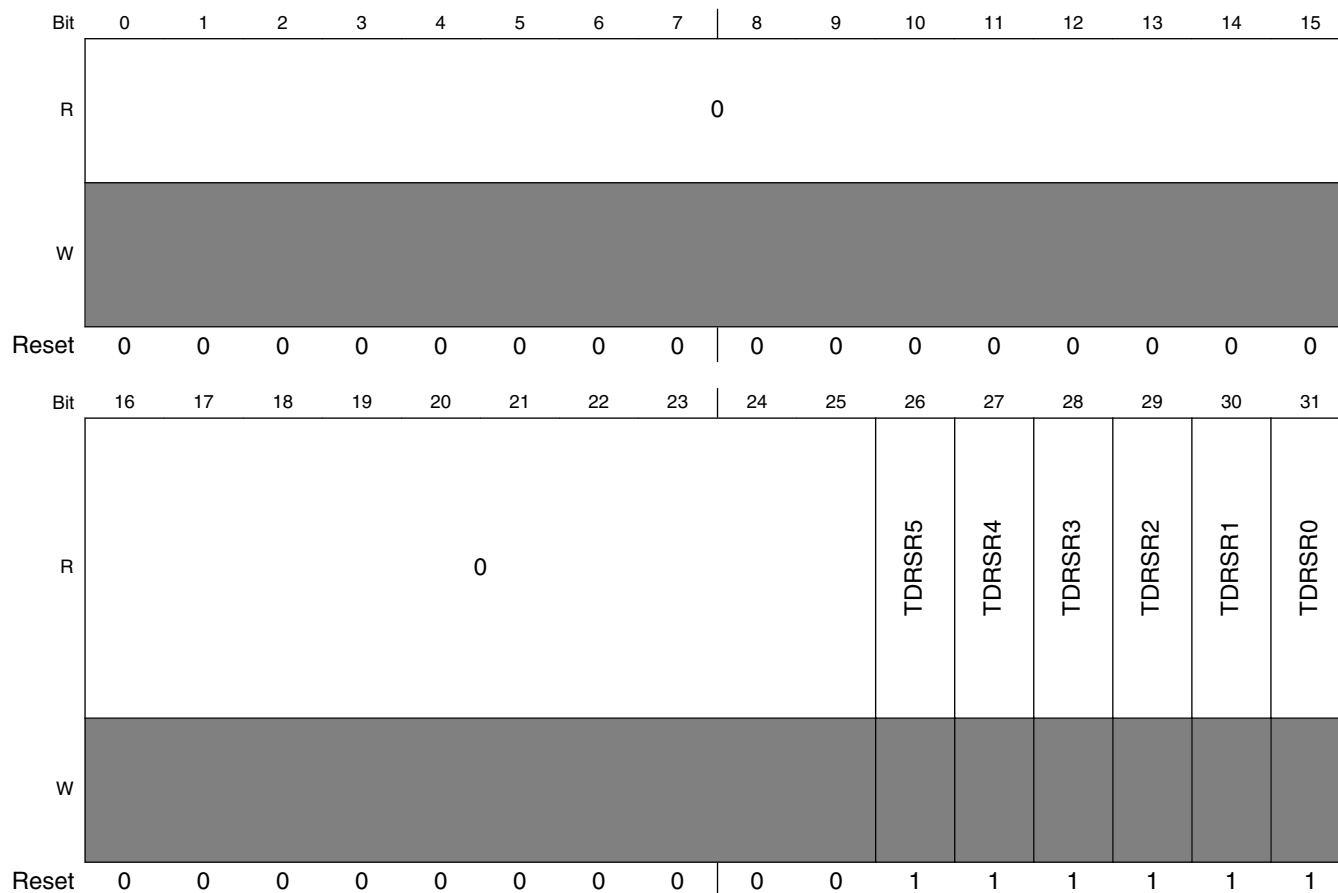
The register contains a status bit for each TDR. Status is determined as follows:

1. The default status for each TDR is '1', which indicates that the erase operation is disabled for all flash blocks associated with the TDR.
2. The status bit for a TDR is set to '0' (indicating that the erase operation for all flash blocks associated with the TDR is enabled) if either a successful programming operation to the TDR diary region has been performed or the TDR Override bit for that TDR is set.

**NOTE**

- TDRSR is a read-only register and writes have no effect on it. Write operations result in a transfer error.
- If the reset value for this register is determined by a DCF record, it will be noted in the chip-specific information at the beginning of this chapter. Otherwise, the reset value is as shown.

Address: 0h base + 0h offset = 0h

**TDM\_TDRSR field descriptions**

| Field            | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–25<br>Reserved | This field is reserved.<br>This read-only field is reserved and always has the value 0.                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 26<br>TDRSR5     | Represents the status of TDR 5 in conjunction with the Diary Override Enable status.<br><br>0 TDR is unlocked. Blocks assigned to this TDR are NOT blocked for erasing by the TDM module. This unlock state is achieved by a successful programming operation to the correct diary region or this TDR has been permanently unlocked by a DCF record that writes to the Tamper Region Override DCF register or a successful software override operation.<br>1 TDR is locked. Erase of blocks assigned to this TDR is not possible. |
| 27<br>TDRSR4     | Represents the status of TDR 4 in conjunction with the Diary Override Enable status.<br><br>0 TDR is unlocked. Blocks assigned to this TDR are NOT blocked for erasing by the TDM module. This unlock state is achieved by a successful programming operation to the correct diary region or this TDR has been permanently unlocked by a DCF record that writes to the Tamper Region Override DCF register or a successful software override operation.<br>1 TDR is locked. Erase of blocks assigned to this TDR is not possible. |
| 28<br>TDRSR3     | Represents the status of TDR 3 in conjunction with the Diary Override Enable status.<br><br>0 TDR is unlocked. Blocks assigned to this TDR are NOT blocked for erasing by the TDM module. This unlock state is achieved by a successful programming operation to the correct diary region or this TDR                                                                                                                                                                                                                             |

*Table continues on the next page...*

**TDM\_TDRSR field descriptions (continued)**

| Field        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|              | has been permanently unlocked by a DCF record that writes to the Tamper Region Override DCF register or a successful software override operation.<br>1 TDR is locked. Erase of blocks assigned to this TDR is not possible.                                                                                                                                                                                                                                                                                                       |
| 29<br>TDRSR2 | Represents the status of TDR 2 in conjunction with the Diary Override Enable status.<br><br>0 TDR is unlocked. Blocks assigned to this TDR are NOT blocked for erasing by the TDM module. This unlock state is achieved by a successful programming operation to the correct diary region or this TDR has been permanently unlocked by a DCF record that writes to the Tamper Region Override DCF register or a successful software override operation.<br>1 TDR is locked. Erase of blocks assigned to this TDR is not possible. |
| 30<br>TDRSR1 | Represents the status of TDR 1 in conjunction with the Diary Override Enable status.<br><br>0 TDR is unlocked. Blocks assigned to this TDR are NOT blocked for erasing by the TDM module. This unlock state is achieved by a successful programming operation to the correct diary region or this TDR has been permanently unlocked by a DCF record that writes to the Tamper Region Override DCF register or a successful software override operation.<br>1 TDR is locked. Erase of blocks assigned to this TDR is not possible. |
| 31<br>TDRSR0 | Represents the status of TDR 0 in conjunction with the Diary Override Enable status.<br><br>0 TDR is unlocked. Blocks assigned to this TDR are NOT blocked for erasing by the TDM module. This unlock state is achieved by a successful programming operation to the correct diary region or this TDR has been permanently unlocked by a DCF record that writes to the Tamper Region Override DCF register or a successful software override operation.<br>1 TDR is locked. Erase of blocks assigned to this TDR is not possible. |

**61.6.2 Last Flash Programmed Address Register (TDM\_LFPAR)**

This register holds information of the address of the last successful programming operation provided from the flash memory to the TDM. Writes to this register are ignored and will generate a transfer error.

Address: 0h base + 4h offset = 4h

|       |       |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0     | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | LFPAR |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |       |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0     | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |

**TDM\_LFPAR field descriptions**

| Field         | Description                                                                                                   |
|---------------|---------------------------------------------------------------------------------------------------------------|
| 0–31<br>LFPAR | Last Flash Programmed Address. These bits represent the address of the last successful programming operation. |



### 61.6.3 Diary Base Address (TDM\_DBA)

This register holds base address information of the diary. Any write to this register is ignored and generates a transfer error.

Address: 0h base + 8h offset = 8h

|       |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0   | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | DBA |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | x*  | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* | x* |    |

\* Notes:

- x = Undefined at reset.

#### TDM\_DBA field descriptions

| Field       | Description                                                                                                                                                                                                                               |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>DBA | Diary Base Address. These bits represent the address of the diary location in the flash.<br><br><b>NOTE:</b> The diary base address is established via the DBA DCF record. The reset value of this register is determined by that record. |

### 61.6.4 Software Tamper Override Key Region (TDM\_STO\_KEYn)

The STO\_KEYn register holds the service key for overriding the Tamper Detect Region n, where n denotes the Tamper Detect Region number.

#### Caution

1. The STO\_KEYn registers can only be written once.
2. The associated DCF client is writable from 0 to 1 only.  
Attempted writes of bits from 1 to 0 are ignored.

Address: 0h base + 10h offset + (4d × i), where i=0d to 5d

|       |         |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|-------|---------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| Bit   | 0       | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| R     | STO_KEY |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| W     |         |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Reset | 0       | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |

#### TDM\_STO\_KEYn field descriptions

| Field           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0–31<br>STO_KEY | Software Tamper Override Key Diary n. This field holds the service key value for overriding Tamper Detect Region n. By loading STO_KEYn with the correct signature of 0x55AA5A5A, Tamper Detect Region n is overridden, and the flash blocks assigned to Tamper Detect Region n are open for erase. This field has no effect when STO_DISn field of STO_DIS DCF client is set.<br><br><b>Caution:</b> The STO_KEYn registers can only be written once. |

**TDM\_STO\_KEY $n$  field descriptions (continued)**

| Field | Description |
|-------|-------------|
|-------|-------------|

## 61.7 Functional description

The primary purpose of the TDM is to provide a Tamper Detection mechanism by enabling an Erase Counter (or diary) for regions within the flash. These regions are called Tamper Detection Regions. The following sections cover the entire functionality of this module.

### 61.7.1 Flash erase counter and tamper detection

A flash region block or part of it is assigned for implementing the diary. The base address for the diary is defined by a DCF Record.

The diary is divided into 6 parts where each part consists of  $256 \times 64$ -bit records, to record up to 256 write attempts on the flash blocks assigned to each Tamper Detection Region. For correct operation, the diary should reside in a flash block that is assigned as OTP.

Each flash block can be assigned to any of the TDRs. Related to each TDR is a group of  $4 \times 32$ -bit internal hardware registers that are initialized by DCF records that define the tamper region. These registers mirror exactly the LOCK registers bits from the flash module.

Each TDR $n$ \_LOCK $m$  register is loaded at boot time by the SSCM, which defines if the associated flash block is Locked for Erase in a particular TDR:

- '1' means the flash block is locked for erase
- '0' means the flash block is not locked for erase

Before performing an erase in one of the blocks locked for erase and associated to a TDR, a program operation has to be performed on the diary part for that TDR. As the diary should be OTP, successive writes are allowed on unprogrammed double words only. This ensures that diary records can only increase and that data recorded in the diary is not modifiable by software.

At the end of a successful double-word program operation, the flash module indicates to the TDM the address of the programmed double word. The TDM determines whether the programmed data was written to the correct TDR diary location and then unlocks the

blocks for erase operation for the corresponding TDR. The TDM ensures that this erase operation on unlocked blocks is allowed until the completion of the erase sequence or until next destructive reset.

When attempting to update the TDM diary and erase flash blocks, set MCR[PECIE] in the Embedded Flash Memory registers to enable interrupt notification for program and erase operations. Then clear MCR[PECIE] to service the interrupt after each program or erase operation. This sequence must be executed before attempting to update the TDM diary, and then again before attempting to erase the unlocked flash blocks.

### NOTE

All blocks associated to a TDR where an erase operation is allowed can be erased all together.

The data programmed in the diary location is flexible, and may include count information or other important customer information.

### NOTE

There is no restriction on the type or format of data that can be programmed in the diary, so software must verify whether the diary is full. If a TDR's diary area is full, the tamper detect mechanism can be disabled for that TDR by creating a Tamper Region Override (TO) DCF record with the TDR's corresponding TDR Override Enable (TOE) bit set to '1'. This allows erase operations to be performed to the TDR without any diary entry.

The setting from “override tamper region lock” DCF record ([Diary](#)) has priority over the settings in the TDR<sub>n</sub>\_LOCK0 – 3 registers. In other words, if a TDR is unlocked for erase by the override tamper region lock, the TDR<sub>n</sub>\_LOCK<sub>m</sub> bits have no effect on the TDR that has been overridden, but the tamper detection remains active for all TDRs not having the override bit set.

Moreover, if a block is locked by the PASS\_LOCK<sub>x</sub>\_PG<sub>n</sub> registers or by TDR<sub>n</sub>\_LOCK<sub>m</sub>, the block is locked for erase.

### NOTE

The settings in the Tamper Region Override Register have priority over the TDR<sub>n</sub>\_LOCK<sub>m</sub> registers. The PASS module also has priority over the settings done in the TDM. For instance, if blocks in a TDR are unlocked for erase operation via the TDM, but the same blocks are programmed to be locked for erase operation in PASS, the blocks end up being locked for erase operation.

Attempts to erase the flash without first writing the diary will be ignored by the flash. The contents in the flash block will remain unchanged and MCR[PEG] will be asserted, indicating the operation was successful and the contents of the block are properly protected from the erase operation.

### 61.7.2 Assigning blocks to Tamper Detect Regions (TDR)

Each TDR is defined by a group of 4 x 32-bit DCF records and clients, with each containing bits that map to specific blocks in a flash region. These mappings mirror exactly the LOCK registers bits from the flash module. Hence, the block assignment registers are named as  $TDR_n\_LOCK_m$  where  $n$  is the number of a tamper region, that is, 0- 5 and  $m$  varies from 0-3 as flash contains four Lock registers.

In these DCF records:

- '1' means that the corresponding flash block is assigned to this TDR
- '0' means that the corresponding flash block is not assigned to this TDR (Default)

DCF clients are not visible to software. They can not be read nor written. They can only be initialized by DCF record within the UTest flash block during reset. The value of these registers can be evaluated by reading all the UTest DCF records and searching for records that correspond to these clients.

#### NOTE

No flash block should be assigned to more than one TDR. Each block should correspond to a single TDR. For example, if a single block is assigned to TDR0 and TDR1, then even if the Diary is programmed for both TDR0 and TDR1, this flash block will never be unlocked for erase operation.

### 61.7.3 Diary

The diary is a region of the flash memory where records of block erases for each TDR are stored. The format and size of the records are not fixed. The only requirements are that each diary has a maximum size of 2 KB, and the minimum size of any programming operation is 8 bytes to honor OTP restrictions. Therefore, each diary can hold a maximum of 256 64-bit entries.

A flash region block or part of it must be assigned for implementing the diary. The base address for the diary is determined by a value written in a DCF record. The diary can be placed within any flash block in flash array, and history records in it can be read by any core.

**NOTE**

To maintain the security of the TDR, the block where the diary is placed must be assigned as OTP within the OTP registers.

The base address of the diary is implemented using a DCF Client. The DCF client is only writable once and the default bit state is 1.

The DCF clients can not be read nor written by software. They can only be initialized by DCF record from within UTEST block during reset. The value of these registers can be evaluated by reading all the UTEST DCF records and searching for records that write these DCF clients.

### 61.7.4 Specifying the diary base address

The diary is a region of flash memory where records of block erases for each TDR are stored. The format and size of these records is defined by the customer. The only hardware constraints are that each diary has a maximum size of 2 KB and the minimum size of any programming operation is 8 bytes. Therefore, each diary region limits the erases of the assigned blocks to 256 erases.

The diary can be placed within any flash block in the flash array. To maintain the security of the TDR, the block where the diary is placed, must be assigned as OTP within the OTP registers, while in customer delivery or OEM production life cycle states.

The base address of the diary is implemented with a 20-bit DCF client that can be initialized by the SSCM during reset. The DCF client is writable once and the default bit state is '1'.

The base address register is not visible to software. It cannot be read nor written. It can only be initialized by DCF record within UTest flash during reset. The value of this register can be evaluated by reading all the UTest flash DCF Records and searching for record that writes this register.

Each TDR is assigned a 2 KB section of the diary. With 6 TDRs, 12 KB of diary space is required.

The diary base address must be at a 16 KB boundary. Therefore the least significant 14 bits of the base address must all be '0'.

The Base address of the diary is represented by:

Diary\_Base\_Address = 0bxxxx\_xxxx\_xxxx\_xxxx\_xx00\_0000\_0000\_0000

The base address of each TDR is shown below:

- TDR0 = Diary\_Base\_Address + 0b00\_0000\_0000\_0000

- $\text{TDR1} = \text{Diary\_Base\_Address} + 0b00\_1000\_0000\_0000$
- $\text{TDR2} = \text{Diary\_Base\_Address} + 0b01\_0000\_0000\_0000$
- $\text{TDR3} = \text{Diary\_Base\_Address} + 0b01\_1000\_0000\_0000$
- $\text{TDR4} = \text{Diary\_Base\_Address} + 0b10\_0000\_0000\_0000$
- $\text{TDR5} = \text{Diary\_Base\_Address} + 0b10\_1000\_0000\_0000$
- End of the diary =  $\text{Diary\_Base\_Address} + 0b11\_0000\_0000\_0000$

### 61.7.5 TDR lock status

The status of whether a TDR is locked for erase is maintained by the TDM. This is accomplished by implementing a register with as many bits as the number of TDRs. The default state of these bits is 1, which is defined as "TDR Erase Blocked". See [TDR Status Register \(TDM\\_TDRSR\)](#) for further details about this register.

When the flash memory module has completed a programming operation, the address of the location programmed is output from the flash module. The TDM in turn compares that address to determine if it falls within the range of any TDR diary regions. If it is within any TDR, the corresponding status bit is set to 0 meaning "TDR Erase Enabled".

These status bits are all reset to 1 when the flash module signals an erase complete by asserting the PEC (Program Erase Complete) signal.

The state of the 6 status bits can be read through the TDRSR. See [TDR Status Register \(TDM\\_TDRSR\)](#). The address of the last successful programming operation provided from the flash memory to the TDM can be read through the LFPAR. See [Last Flash Programmed Address Register \(TDM\\_LFPAR\)](#).

### 61.7.6 TDR override

Before a block protected by a tamper detect region can be erased, a record must be written to the associated diary. Software must search through the diary to find the end of the records, where the next available diary location that can be written. After a maximum of 256 records are programmed to the diary, a TDR diary is full. Full means that at least 1 bit has been programmed to '0' in each 64 bit double word within the 2 KB diary.

If software determines that a particular TDR diary is full it may override the diary to allow erasing to continue or it may choose to stop further erases and to flag an error.

Diary Override bits are implemented with a DCF client consisting of 6 bits, one for each TDR. See the device configuration information at the beginning of this chapter for TDM DCF client details.

The Diary Override bits are by default set to '0'; which means the diary must be written before an erase can occur. The DCF client can be written many times but the bits can only be written to '1'. Writes to '0' are ignored. In this way, by default, all Tamper Detect Regions are active. The Tamper detect region diaries can be individually overridden by writing a DCF record that sets that corresponding bit to '1'. In this mode, the TDR status bit is permanently set to 0 to allow flash erase.

The application can also override the diary by loading a service key to the `STO_KEYn` program-visible register, where *n* denotes the tamper detect region. By loading the value `0x55AA5A5A` into `STO_KEYn` register, the corresponding tamper detect region is overridden. This software override mechanism is disabled if the `STO_DISn` field of the `STO_DIS` DCF Client is set.

## 61.7.7 One time programmable (OTP)

### 61.7.7.1 Overview

Any flash block within the flash array can be assigned, at any time, to be OTP. Once a flash block is assigned OTP, it cannot be changed back.

OTP means that flash erase of the entire block is disabled and only 64-bit double words that are already erased (that is, flash content is `0xFFFF_FFFF_FFFF_FFFF`) can be programmed. Over-programming is not possible.

Flash blocks are assigned as OTP by writing a DCF record. The block will become OTP after the next functional reset.





**How to Reach Us:****Home Page:**[nxp.com](http://nxp.com)**Web Support:**[nxp.com/support](http://nxp.com/support)

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: [nxp.com/SalesTermsandConditions](http://nxp.com/SalesTermsandConditions).

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer's applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, Altivec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, and UMEMS are trademarks of NXP B.V. All other product or service names are the property of their respective owners. Arm, AMBA, Artisan, Cortex, Jazelle, Keil, SecurCore, Thumb, TrustZone, and  $\mu$ Vision are registered trademarks of Arm Limited (or its subsidiaries) in the EU and/or elsewhere. Arm7, Arm9, Arm11, big.LITTLE, CoreLink, CoreSight, DesignStart, Mali, Mbed, NEON, POP, Sensinode, Socrates, ULINK and Versatile are trademarks of Arm Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© 2013–2018 NXP B.V.